Accurate Modeling of Stability & Control Properties for Fighter Aircraft from CFD

THESIS

Jedediah H. Butler, Civ, USAF

AFIT/GAE/ENY/12-M04

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the United States Government. This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

AFIT/GAE/ENY/12-M04

Accurate Modeling of Stability & Control Properties for Fighter Aircraft from CFD

THESIS

Presented to the Faculty
Department of Aeronautics and Astronautics
Air Force Institute of Technology
Air University
Air Education and Training Command
In Partial Fulfillment of the Requirements for the
Degree of Master of Science in Aeronautical Engineering

Jedediah H. Butler, BS
Civ, USAF

March 2012

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

AFIT/GAE/ENY/12-M04

ACCURATE MODELING OF STABILITY & CONTROL PROPERTIES FOR FIGHTER AIRCRAFT FROM CFD

Jedediah H. Butler, BS
Civ, USAF

Approved:

3/12/12
Maj A. Lofthouse, PhD (Chairman)
date

3/8/12
D. McDaniel, PhD (Member)
date

Donald L. Kunz
9 March 2012
D. Kunz, PhD (Member)
date

AFIT/GAE/ENY/12-M04

Abstract

Difficulties in obtaining accurate Stability and Control (S&C) data for nonlinear regions in the flight envelope early in the design phase often lead to costly fixes late in the acquisition process. Work examined herein addresses this problem by utilizing recent advances in Computational Fluid Dynamics (CFD) to create a high-fidelity database before any parts have to be physically produced. In combination with CFD, System Identification (SID) is used to characterize the S&C characteristics of aircraft by some Reduced Order Model (ROM). The method of obtaining enough data in the "right" places, via some optimized maneuver, to create an accurate ROM is the main focus of this work. A series of metrics were created and validated to help identify the quality of any maneuver before it is even run. To validate the metrics, seven different maneuvers were generated, run, and used to create models. These models are then evaluated against sets of results from validation maneuvers to show accuracy of modeling a given regressor space - the "regressor space" being the required variations of the model variables (angle of attack, pitch rate, etc.). From the validated metrics, the "best" maneuver for creating the most accurate model of a given grid within a specified regressor space is defined.

Acknowledgements

First and foremost, I am thankful to God for allowing me this opportunity to study a topic that I enjoy under Maj. Lofthouse and Dr. McDaniel. Also, I could not complete such a task without the support, hard work, and understanding of my wife, son, and my family. I would also like to thank the High Performance Computing Center (HPCC) for the many hours of computational time given on the Raptor system. Without the use of such a system, the level of CFD analysis accomplished could not have been realized.

Jedediah H. Butler

Table of Contents

Page

Abstract iv

Acknowledgements v

List of Figures ix

List of Tables xiv

List of Abbreviations xv

I. Introduction 1
  1.1 General Issue 1
  1.2 Background 2
    1.2.1 Wind Tunnel 2
    1.2.2 Flight Test 3
    1.2.3 CFD 3
    1.2.4 Modeling Data 4
  1.3 Research Focus 5
  1.4 Research Objectives 6
  1.5 Road Map 7

II. Background 9
  2.1 Flow Physics 9
    2.1.1 Expected Flow Features 9
    2.1.2 Separation 11
    2.1.3 Vorticity and Q-Threshold 12
    2.1.4 Boundary Layer Theory 15
    2.1.5 Turbulence Theory 18
    2.1.6 Turbulence Modeling 23
  2.2 CFD Settings 26
    2.2.1 Grid Development 27
    2.2.2 Flow Solver and Settings 28
  2.3 Stability Derivatives 30
  2.4 System Identification 34
    2.4.1 SID Techniques 35
    2.4.2 System Identification Program for Aircraft . . . 36
  2.5 Previous Research 37

Page

2.5.1 Initial Attempts 37
2.5.2 Current State-of-the-Art 38
2.5.3 Where This Research Fits In 39

III. Methodology 41
3.1 Overview 41
3.2 Refining the Problem 41
3.3 Metric Development 42
3.3.1 The Metrics 43
3.4 Maneuver Development 48
3.4.1 The Final Maneuvers 50
3.4.2 Outputs from Metrics 68
3.4.3 Comparison Maneuvers 77
3.4.4 Hypothesis 82
3.5 Final Solver Settings 83
3.5.1 Computational Resources 85
3.6 Grid Refinement 86
3.6.1 High AoA Refinement 88
3.6.2 Low AoA Refinement 96

IV. Results 106
4.1 Results of Maneuvers 106
4.2 Created Models 107
4.3 Prediction of Comparison Results 109
4.3.1 Mach 0.5 109
4.3.2 Mach 0.9 125

V. Conclusions 132
5.1 Summary 132
5.2 Recommendations 134
5.2.1 Future Work 134

Appendix A. `Matlab`® Code for Metric Calculation 136

Appendix B. Complete List of Solver Settings and Boundary Conditions 142

Appendix C. `Matlab`® Code for Maneuver Development 145

Appendix D. `Matlab`® Code for Creating Arbitrary Motion File . . . 162

Appendix E. `Matlab`® Code for Creating Outputs of Models 163

Page

Bibliography 175

Vita 179

List of Figures

Figure Page

2.1. Depiction of drag profile as Mach number increases from subsonic to supersonic speeds [6]. 10

2.2. Example of strake on aircraft [5]. 11

2.3. Effects of increased AoA on flow over airfoil 12

2.4. Airflow from bottom to top of wing [42]. 14

2.5. Vorticity produced at tip of wing [42]. 14

2.6. Boundary layer regions [41] 17

2.7. View of turbulent structures on plate [41]. 19

2.8. Energy production and dissipation in turbulent flow [41]. . . . 19

2.9. Comparison of very fine grid (top) to grid created with AMR (bottom) [13] 28

2.10. Comparison of results of successively refined grids (top), fine grid (bottom left), and AMR grid (bottom right) [13]. 29

2.11. Depiction of Static stability, neutrality, and instability [2] . . . 31

2.12. Slope of pitching moment for static stability, neutrality, and instability [1] 32

2.13. Depiction of dynamic stability, neutrality, and instability [2] . . 33

2.14. Block diagram of relationship of system to known inputs and outputs. 34

3.1. Example of the discretized regressor space. 43

3.2. Sample arbitrary motion input file [12] 51

3.3. Motion of first maneuver. 52

3.4. Regressor space covered by the first maneuver. 52

3.5. Pitch rate of first maneuver compared to AoA. 53

3.6. Motion of second maneuver. 54

3.7. Regressor space covered by the second maneuver. 54

3.8. Pitch rate of second maneuver compared to AoA. 55

3.9. Initial motion of third maneuver before and after "chopping." 56

3.10. Corners of third maneuver before and after smoothing sharp corners from being chopped 57

3.11. Motion of third maneuver. 58

3.12. Regressor space covered by the third maneuver. 58

3.13. Pitch rate of third maneuver compared to AoA. 59

3.14. Fourth maneuver. 61

3.15. Regressor space covered by the fourth maneuver. 62

3.16. Pitch rate of fourth maneuver compared to AoA. 62

3.17. Fifth maneuver. 63

3.18. Regressor space covered by the fifth maneuver. 63

3.19. Pitch rate of fifth maneuver compared to AoA. 64

3.20. Motion of sixth maneuver. 67

3.21. Regressor space covered by the sixth maneuver. 67

3.22. Pitch rate of sixth maneuver compared to AoA. 68

3.23. Motion of seventh maneuver. 69

3.24. Regressor space covered by the seventh maneuver. 69

3.25. Pitch rate of seventy maneuver compared to AoA. 70

3.26. Discretized regressor space for first maneuver 71

3.27. Discretized regressor space for second maneuver 71

3.28. Discretized regressor space for third maneuver 72

3.29. Discretized regressor space for fourth maneuver 72

3.30. Discretized regressor space for fifth maneuver 73

3.31. Discretized regressor space for sixth maneuver 73

3.32. Discretized regressor space for seventh maneuver 74

3.33. Motion of first comparison maneuver. 78

3.34. Regressor space covered by the first comparison maneuver. . . . 79

Figure Page

3.35. Motion of second comparison maneuver. 80

3.36. Regressor space covered by the second comparison maneuver. . 81

3.37. Motion of third comparison maneuver. 81

3.38. Regressor space covered by the third comparison maneuver. . . 82

3.39. Views of the original grid. 86

3.40. Flow solution of initial grid at Mach 0.5 and AoA = 30. 89

3.41. Flow solution of initial grid at Mach 0.9 and AoA = 30. 89

3.42. Flow solution of initial grid at Mach 0.5 and AoA = 30. 90

3.43. Flow solution of initial grid at Mach 0.9 and AoA = 30. 90

3.44. First refined region for AoA = 30 *deg* in X direction. 91

3.45. First refined region for AoA = 30 *deg* in Y direction. 92

3.46. First refined region for AoA = 30 *deg* in Z direction. 92

3.47. Second refined region for AoA = 30 *deg* in X direction. 93

3.48. Second refined region for AoA = 30 *deg* in Y direction. 93

3.49. Second refined region for AoA = 30 *deg* in Z direction. 94

3.50. Grid refinement results for lift at 30 *deg* AoA. 94

3.51. Grid refinement results for drag at 30 *deg* AoA. 95

3.52. Grid refinement results for pitching moment at 30 *deg* AoA. . 95

3.53. Flow solution of grid at Mach 0.5 and AoA = -10. 96

3.54. Flow solution of grid at Mach 0.9 and AoA = -10. 97

3.55. Flow solution of grid at Mach 0.5 and AoA = -10. 98

3.56. Flow solution of grid at Mach 0.9 and AoA = -10. 98

3.57. First refined region for AoA = -10 *deg* in X direction. 99

3.58. First refined region for AoA = -10 *deg* in Y direction. 99

3.59. First refined region for AoA = -10 *deg* in Z direction. 100

3.60. Second refined region for AoA = -10 *deg* in X direction. 101

3.61. Second refined region for AoA = -10 *deg* in Y direction. 101

3.62. Second refined region for AoA = -10 *deg* in Z direction. 102

3.63. Third refined region for AoA $= -10$ *deg* in X direction. 102

3.64. Third refined region for AoA $= -10$ *deg* in Y direction. 103

3.65. Third refined region for AoA $= -10$ *deg* in Z direction. 103

3.66. Grid refinement results for lift at -10 *deg* AoA. 104

3.67. Grid refinement results for drag at -10 *deg* AoA. 104

3.68. Grid refinement results for pitching moment -10 *deg* AoA. . . 105

4.1. Kestrel user interface for creation of models 107

4.2. All models with maximum order of 2 predicting lift for CM1. . 110

4.3. All models with maximum order of 3 predicting lift for CM1. . 110

4.4. Zoomed in peak region of model predictions for CM1. 111

4.5. Zoomed in static region of model predictions for CM1. 112

4.6. All models with maximum order of 2 predicting drag for CM1. 112

4.7. All models with maximum order of 3 predicting drag for CM1. 113

4.8. All models with maximum order of 2 predicting pitching moment for CM1. 113

4.9. All models with maximum order of 3 predicting pitching moment for CM1. 114

4.10. Flow field of CM1 near zero pitch rate at -10 *deg* AoA. 115

4.11. Flow field of static solution at $m10$ *deg* AoA. 115

4.12. Flow field of CM1 at high, positive pitch rate 8 *deg* AoA. . . . 116

4.13. Flow field of static solution at 8 *deg* AoA. 116

4.14. All models with maximum order of 2 predicting lift for CM2. . 117

4.15. All models with maximum order of 3 predicting lift for CM2. . 118

4.16. All models with maximum order of 2 predicting drag for CM2. 118

4.17. All models with maximum order of 3 predicting drag for CM2. 119

4.18. All models with maximum order of 2 predicting pitching moment for CM2. 119

4.19. All models with maximum order of 3 predicting pitching moment for CM2. 120

4.20. Comparison of all models predicting static C_L versus AoA curve. 120
4.21. Comparison of all models predicting static C_D versus AoA curve. 121
4.22. Comparison of all models predicting static C_m versus AoA curve. 121
4.23. All models with maximum order of 3 predicting lift for CM3. . 126
4.24. All models with maximum order of 3 predicting drag for CM3. 127
4.25. All models with maximum order of 3 predicting pitching moment for CM3. 128
4.26. Comparison of all models predicting static C_L versus AoA curve. 129
4.27. Comparison of all models predicting static C_D versus AoA curve. 130
4.28. Comparison of all models predicting static C_m versus AoA curve. 130

List of Tables

Table Page

3.1. List of Inputs to Eq. (3.17) for each maneuver. 51

3.2. Output of first seven metrics for all seven maneuvers. 75

3.3. Output of last six metrics for 5 $\frac{deg}{s}$ cutoff between high and low Q. 75

3.4. Output of last six metrics for 10 $\frac{deg}{s}$ cutoff between high and low Q. 75

3.5. Output of last six metrics for 20 $\frac{deg}{s}$ cutoff between high and low Q. 76

3.6. Output of last six metrics for 50 $\frac{deg}{s}$ cutoff between high and low Q. 76

3.7. List of Inputs to Eq. (3.17) for each comparison maneuver. . . 79

4.1. Summary of finished versus unfinished cases. 106

4.2. Summary of computational expense. Total number of hours ran = 1.64 million 106

4.3. R^2 metrics for each model with maximum term order of two. . 123

4.4. R^2 metrics for each model with maximum term order of three. 123

4.5. R^2 metrics for each model with maximum term order of two. . 125

4.6. R^2 metrics for each model with maximum term order of three. 125

4.7. R^2 metrics for each model with maximum term order of four. . 126

B.1. kAVUS Inputs 142

List of Abbreviations

Abbreviation		Page
S&C	Stability and Control	1
CFD	Computational Fluid Dynamics	2
6DOF	6 Degree of Freedom	3
SID	System Identification	4
ROM	Reduced Order Model	5
AoA	Angle of Attack	9
N-S	Navier-Stokes	20
RANS	Reynolds Averaged Navier-Stokes	22
DNS	Direct Numerical Simulation	23
LES	Large-Eddy Simulation	25
DES	Detached Eddy Simulation	25
DDES	Delayed DES	26
AMR	Adaptive Mesh Refinement	27
AVUS	Air Vehicles Unstructured Solver	28
kAVUS	Kestrel AVUS	29
SIDPAC	System Identification Program for Aircraft	36
COMSAC	Computational Methods for Stability and Control	37
SRC	Supercomputing Resource Center	85
TM	Training Maneuver	106
CM	Comparison Maneuver	106

ACCURATE MODELING OF STABILITY & CONTROL PROPERTIES FOR FIGHTER AIRCRAFT FROM CFD

I. Introduction

1.1 General Issue

Inaccurate predictability of Stability and Control (S&C) characteristics of aircraft early in the design phase has been the source of many costly design changes late in the development of military fighters for decades. High-fidelity results of "simple" configurations are easy to come by via simple linear methods, whereas difficulties arise in nonlinear regions of the flight envelope. The often large error in S&C prediction of a high-performance aircraft heading into the flight test phase of development results in precious flight test hours being spent in benign conditions for safety reasons. Then as the flight envelope is incrementally widened to include more nonlinear regimes, these errors in prediction generally reveal themselves in the form of costly fixes that have to be made.

At this late stage of the acquisition cycle, the initial flight test prototypes have obviously been produced whose entire flight control system has been based on initial bare airframe S&C data available at the time. Problems can also arise where the structure itself has to be modified as well. This can be due to higher than expected loading, improper control surface sizing, or a number of other issues. One common problem, for example, as stated by William Thomas and William Blake, is that over the last 50 years many "fighters suffered from roll coupling brought on by a combination of high rates of roll and insufficient stability to counter the resultant buildup in alpha and beta. Enlarging the vertical tails was the most common and successful solution" [20].

The issues stated above are only a few examples of how determining a more accurate representation of the aircraft early in the design phase will save time and money. Time by helping to keep a program on track and money by saving potentially billions of dollars [25] in costly *ad hoc* fixes that are applied without a sound basis of the fundamental physics concerned. These problems will only get worse with more complex aircraft that are sure to be seen in the future [21].

1.2 Background

It is well known that today there are four main avenues of obtaining aerodynamic data for an aircraft: Analytical analysis, wind tunnel testing, flight testing, and Computational Fluid Dynamics (CFD). Analytical analysis includes linear aerodynamic techniques, sets of data sheets, empirical relations, etc that can produce results in as little as minutes. This method is useful due to its simplicity, fast turn-around time, and ability to give a "starting-point" in design. The results compare well to benign flight conditions but basic assumptions utilized in creating these methods fall apart outside of such scenarios.

1.2.1 Wind Tunnel. Wind tunnel testing is often used extensively due to the ability to use a representative model in an actual fluid to obtain results. The accuracy of these results, even in nonlinear flight regions, can be quite accurate if done properly. Wind tunnel tests must be done in such a manner that sting effects are removed, wall effects are minimized, instrumentation is accurately placed and calibrated, etc. The results of the wind tunnel test must then be properly filtered and scaled to the proper size while taking Reynolds number effects into account as best as possible. Wind tunnel test can be hampered by availability of a needed wind tunnel itself that meets the desired velocity for a given flight condition and scale of the model. Even then, there are limitations to the dynamic motions that can be conducted in a wind tunnel which are key to understanding dynamic stability derivatives of an aircraft. The entire process of wind tunnel testing is also expensive and time consuming and must be redone as changes are made to the geometry of the aircraft.

1.2.2 Flight Test. Flight testing is by far the most accurate as it is the actual data of the actual aircraft at the actual flight conditions. This is the data that, if available, other methods are trying to show comparison to so the benefits are inherent. The drawbacks, however, are of great concern. First of all, flight test cannot happen until there is first an aircraft to flight test which is not until late in the design phase where millions if not billions of dollars have already been invested. It is the most expensive method of all data gathering processes and is the most time-consuming as it takes time to instrument the aircraft, get approval to fly within a given flight envelope, acquire the data, analyze it, and determine the safety of pressing forward. Even still, there are limitations to the data gathered from flight test. First, the airplane is constrained to maneuvers that are physically flyable (the importance of this will be detailed later) and the desired trajectory may not be met exactly due to error from any number of sources such as wind gusts. Secondly, there is no way to capture what the flow is doing around the body of the aircraft (off-body effects), only the affect of the flow on the aircraft. An example of this is visualizing exactly where the vorticies from wingtips or strakes are going and what they might affect downstream. This second limitation is not so significant for the purpose of this study but is worth stating in general.

1.2.3 CFD. Computational Fluid Dynamics is a relatively new tool that has reached a level of maturity requisite for robust computations. The last nine to ten years have particularly made significant progress in predicting full aircraft Navier-Stokes solutions at not only static points, but also prescribed maneuvers and 6 Degree of Freedom (6DOF) aircraft response [39]. This advancement in prescribed maneuvers, specifically, are critical to determining the dynamic derivatives as they, by virtue of being dynamic, require a time dependent motion to be computed. CFD also allows for tight control over the motion being run as sources of error from flight test are removed. The maneuvers are also not limited to flyable maneuvers where certain aerodynamic parameters may be lumped together. Instead, any type of aircraft motion is possible in the flow field [21]. The only limitations to the maneuvers in CFD are from concerns about the stability of the solution being computed and computational time needed to run it. In general, the overall expense to gathering CFD data is less than wind tunnel and the meshes used to compute the solution are fairly adaptable as the design changes [14].

There are other limitations to CFD, of course. These include limitations in modeling turbulence and transition of the flow. The solution can also be highly dependent on the procedure and settings chosen. Other concerns include: Is the solution grid and time step converged? What level of numeric temporal damping is used for the final solution? Is the solution 1st or 2nd order accurate? What limiter is used? These questions will be detailed at greater length in future chapters, but speaking generally about CFD; turbulence models have made great strides and over time "best practices" have been created, processor power has increased drastically, and results to high-fidelity CFD have been shown to compare well with wind tunnel and flight test.

1.2.4 Modeling Data. Up until now, only the means of obtaining aerodynamic data has been discussed. Once the data is gathered (by whatever means), there are any number of ways to compile this data for useful analysis at a later time. This S&C data can be transformed into polynomials, graphs, table look-ups, or neural networks. Polynomials are good due to their smoothness, but discontinuities in data may require a partitioning approach. Graphs are good at looking at qualitative trends but are not suitable for quantitative analysis. Table look-ups require interpolation to be of practical use and often data in the table must be generated from numerical differentiation which can introduce significant noise. Neural networks approximate any nonlinear function to an arbitrary degree of accuracy but but do not give insight into the aircraft dynamics [21].

System Identification (SID) is another method of compiling the retrieved data by constructing a mathematical model of the system itself given the input and output to the system. SID is similar to two other general problems in aircraft dynamics and control. The first being Simulation, where the output is determined given the input and the system. Second is Control where, in short, the input must be found given the system and the output [26]. SID has been utilized to evaluate wind tunnel and flight test data to obtain an accurate Reduced Order Model (ROM) of an aircraft for many years. This model can then be utilized to simulate the behavior of the aircraft at a different condition than those used to create the model or used in aiding the design of the control system for the aircraft. This evaluation of the dynamic response of the airplane can also be used for analysis of the handling qualities of the aircraft. An accurate analysis of handling qualities in nonlinear flight regions is of significant interest today [42].

1.3 Research Focus

A high-quality mathematical model of an aircraft proves to be an invaluable tool in the design process, including more accurately understanding the response of the system to predict the expected loads during flight, design the control system, and define the stability and controllability of the aircraft due to the sizing of various surfaces, etc. Usually such a model is not available until late in the design phase of an aircraft and even late in the design phase there are often large gaps in data due to budget cuts to flight testing and limitations to maneuvers that can be performed in a wind tunnel.

This is where CFD can play an important role. It can be well utilized to help create a database of S&C data at relatively low cost, with very few “holes,” early in the design process. This data can be fed into current SID tools that have been developed specifically for modeling aircraft. This model can then be continually updated over time as the design is updated to feed into successively higher design elements. Even after the aircraft is produced, CFD can be utilized to help fill in the holes from flight and wind tunnel test.

Dynamic CFD simulations are needed to gain the benefit of enabling a model to be created for dynamic stability derivatives as well as static. These dynamic maneuvers also feed the required “input” into a SID program along with the “output” from which the model can be created to describe the relationship between the two - assuming the aircraft and its aerodynamics have been properly excited by the maneuver. A couple of primary questions fall out from this.

- What maneuver will best excite the full range of aircraft dynamics?
- Is there a way to know how good a maneuver is *before* the computational resources are expended?

The answers to these questions will be the focus of the work included herein.

1.4 Research Objectives

It is important to first define how a maneuver is “good” and in order to do this the criteria must be set. The simplest method to determine the goodness of a model is to quantify the error of model prediction to another set (or sets) of data which lies within the same regressor space as the data used to create the mathematical model. The “regressor space” is the range of values of each of the model input variables (angle of attack, pitch rate, etc.). The other set (or sets) of data must have known inputs and outputs for comparison and accurately cover the desired regressor space trying to be modeled.

The first objective is to tackle the question of whether it is possible to detect how good a maneuver is at modeling a given regressor space before it is run. A set of metrics will need to be developed to provide a basis for how good a maneuver is. These metrics will need to be validated by running multiple maneuvers, each of which either create the best model or maximize a certain attribute of the metrics for validation. These maneuvers will then need to be compared against known data to see which maneuver (or type of maneuver) produces the best results.

The second primary objective is to create maneuvers that have not been utilized in previous research that produce better results. These new training maneuvers can be based on the validated metrics and even used to help validate the metrics. The end goal of this objective is to have a new "best" maneuver.

1.5 Road Map

The first step in this project is to examine what work has already been done in this area. Currently, the best maneuver is a "chirp" sinusoidal motion - a sinusoidal change in angle of attack with varying frequency and even varying amplitude. Only the longitudinal response of an aircraft will be examined as well as it simplifies the maneuver development process and conclusions from longitudinal motion can be translated into the other two axes at a future time.

Two flight conditions will be examined, one subsonic and the other transonic. A regressor space which covers both linear and nonlinear angles of attack will be utilized along with a wide range of pitch rates. A grid of the F-16C will be provided for this research and the newly developed Kestrel Unstructured Air Vehicles CFD solver will be utilized as it includes many features which allow for easy use for grid motion as well as to provide desired testing for the software.

The basis for creating metrics will also be developed by utilizing the input regressor variables and describing how well a given maneuver covers various regions of the regressor space in attempts to provide insight into how good a maneuver is before it is run. For the validation of these metrics, many different maneuvers will be produced along with the current "best" maneuver. The primary downfall of this maneuver lies in its prediction of static data. Therefore, the chirp sinusoidal motion will be modified to include static regions to help increase the prediction of this static data. These maneuvers will then be utilized to create models of the lift, drag, and pitching moment coefficients for the aircraft. These three coefficients are the only ones that need to be modeled for longitudinal motion as any of the dynamic derivatives required for analysis (such as $C_{m,a}$, $C_{m,q}$, etc) can be obtained by taking the respective derivatives of the original three models. These models are then used to predict the results of same coefficients of different maneuvers (deemed comparison maneuvers) which cover various regions of the regressor space in a dissimilar manner than the maneuvers from which models are created and at least one of which will primarily include static data.

The predictions of the various models to the comparison data will likely result in one (or a couple of) maneuver(s) which are better than the rest. This model is then deemed the new (or still standing) best maneuver. From this, the metrics which best describe the characteristics of the best maneuver can be considered to be the most important, thus validating those metrics. In this way, both primary goals are inter-related: The metrics cannot be validated until it is known which maneuvers are best and that is not known until they are all compared to validation data (the comparison maneuvers).

II. Background

2.1 Flow Physics

It is a good practice to first examine what type of flow features are expected before starting a CFD project. Having a good "intuition" of what is going to have to be calculated by the CFD solver will affect the entire process from how the grid is made, what solver and settings should be used, etc. Those details will be examined in the next section, so the current section will guide through what to expect as well as some theory needed to set up later discussion.

2.1.1 Expected Flow Features. This "intuition" does not have to be inherent by the user. Instead, it can be gained by researching similar work to see what results were found. A great initial source to turn to is an aerodynamics textbook. Even though most techniques utilized (at least initially) in these books rely on a linearization of the governing equations and assume inviscid and irrotational flow, a great deal of insight can be gained about the overall flow features to expect. Some considerations when deciding what to research include the flight conditions, aircraft orientation, and types of unique geometry that are on the aircraft. Examples include: What Mach number will be flown? What range of Angles of Attack (AoA), sideslip, etc will be examined? Are the wings swept? Are there strakes? Etc. One last concern is to determine what needs to be modeled? Should the solution be inviscid or viscous? How should turbulence effects be modeled?

The flight conditions can have a drastic effect on a CFD solution as it can restrict what type of flow solver can be used, especially if hypersonic flight is desired. For this research two Mach numbers will be examined. One will be in the transonic region and the other in subsonic. The transonic case will induce important phenomena which need to be discussed.

When an aircraft flies faster than its critical Mach number but before the flow everywhere around the aircraft is above Mach 1, it is defined to be flying within its transonic envelope. This region is roughly in the range of freestream Mach numbers of 0.8 to 1.2 but is highly dependent on the geometry of any aircraft. In this region shocks are formed and drag is increased as can be seen in Figure 2.1. Transonic airflow is also a highly nonlinear region where theoretical aerodynamics tend to fall short in accurate prediction [7]. As stated in Section 1.1, the prediction of these nonlinear aerodynamics is one of the primary reasons for the need of this research. The feasibility of CFD computing accurate results for such nonlinear regions will be discussed in Section 2.2.

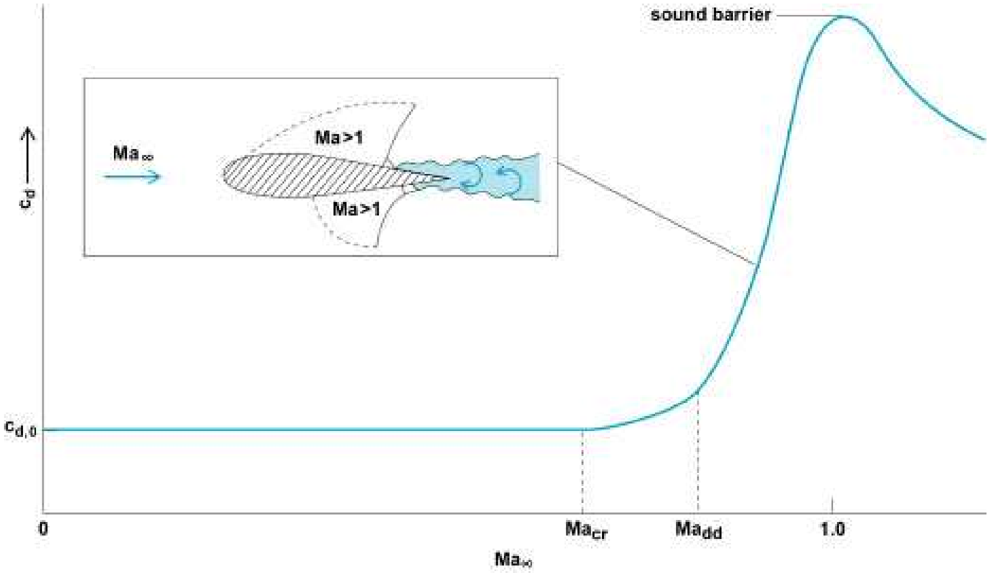


Figure 2.1: Depiction of drag profile as Mach number increases from subsonic to supersonic speeds [6].

The F-16C aircraft to be used is designed for high-speed, high-performance flight. Due to such, it is designed with a thin cropped delta-wing, strakes, and has a relatively small profile. The primary factor which results in the implementation of a swept/delta wing in a design is for transonic/supersonic flight consideration. The swept wing effectively increases the critical Mach number where the flow over the wing becomes greater than Mach 1 even though the freestream Mach is less than 1. This is done by decreasing the speed of the flow perpendicular to the leading edge of the wing [7]. Strakes are essentially very small aspect ratio “wings” with very high sweep, stationed in front of the main wing (see Figure 2.2). The primary purpose of

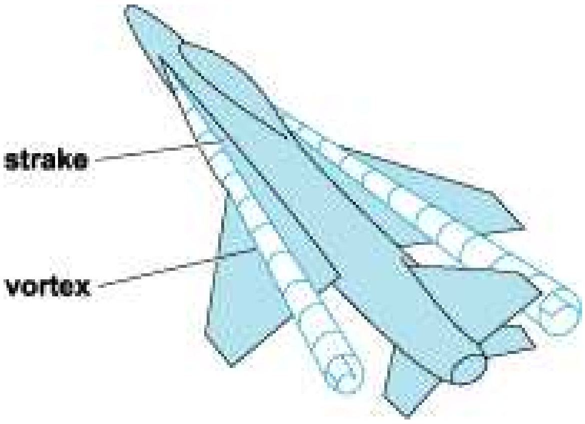


Figure 2.2: Example of strake on aircraft [5].

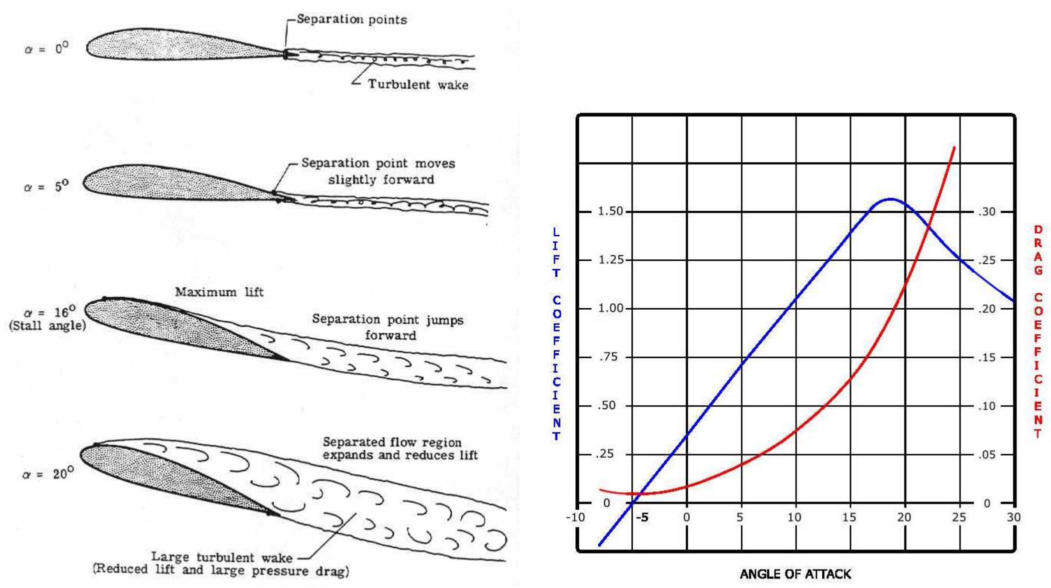
strakes is to produce vorticies which stay attached to the wing surface at high AoA to prolong stall and promote stability [42]. What vorticies are, how they are produced, and techniques for quantifying them will be discussed in Section 2.1.3.

Aircraft orientation will greatly impact the types of flow features seen. The maneuvers considered will vary over a wide range of AoA. This range is from -10 degrees to 30 degrees. It is known that when a wing gets to a certain “high” AoA, flow separation will occur [42]. The magnitude of the AoA where separation occurs varies greatly based on the shape of the wing. When this separation occurs, the wing loses aerodynamic lift and is stalled. There are ways to prolong stall for the entire aircraft when other features, such as strakes, on the aircraft affect the flow to keep it attached to the wing longer.

The last question posed asks what is trying to be modeled about the aircraft? In order to obtain the most accurate results as compared to flight test, it is necessary to model the full turbulent Navier-Stokes equations. In summary of this section, separation, vorticity, and turbulence theory all need to be looked at to understand how they will affect a CFD simulation.

2.1.2 Separation.

Boundary layer separation has critical effects on the flight mechanics of an aircraft. This separation happens as AoA is increased from zero. At low AoA, the boundary layer stays attached but as the AoA increases the coefficient of lift increases linearly until the adverse pressure gradient on the aft portion of the upper surface becomes large enough to cause the boundary layer separation point to "jump" forward (Figure 2.3(a)) and the increase of lift with AoA will lessen (Figure 2.3(b)). As AoA increases more, the complete separation occurs and the lift rapidly decreases. This separation also greatly increases the drag [42]. This nonlinear separated flow region produces large vorticies ([42])in this area above the wing which has a significant dependency on the creation of the grid which will be mentioned in Section 2.2.1 as well as a significant dependency on the turbulence model chosen (Section 2.1.6).



(a) Separation [3]

(b) Lift and drag coefficients [4]

Figure 2.3: Effects of increased AoA on flow over airfoil

2.1.3 Vorticity and Q-Threshold. Vorticity is essentially rotation in a flow field. This rotation is introduced primarily from viscosity which requires fluid velocity exactly on the skin of the aircraft to be zero with respect to the velocity of the aircraft itself. This leads into boundary layer and turbulence theory and will be discussed in Section 2.1.5. The zero-velocity requirement at the surface and non-zero velocity above the surface causes a viscous shear in the flow which causes a rotation [42].

The viscous shear in the three-dimensional velocity field, $U(x,t)$, can be described by the rate-of-strain tensor in Eq. (2.1). Similarly, the rate-of-rotation tensor can be described by Eq. (2.2) [43]. These equations combine to describe the entire velocity gradient of the flow.

$$S_{ij} = \frac{1}{2}\left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i}\right) \tag{2.1}$$

$$\Omega_{ij} = \frac{1}{2}\left(\frac{\partial U_i}{\partial x_j} - \frac{\partial U_j}{\partial x_i}\right) \tag{2.2}$$

Where $x_{i/j}$ is the three dimensional spacial variable.

Haller [22] discusses the difficulty in setting a universal method of defining vorticity that works for any reference frame and problem conditions. The purpose of this section is not to give an overview of many of the posed solutions. One important method, however, needs to be mentioned as it is a helpful criterion utilized in many flow solvers, including Kestrel. It is the Q-criterion or Q-threshold criterion. This criterion attempts to produce a Galilean invariant vortex criterion which defines a vortex as a spatial region where:

$$Q = \frac{1}{2}\left[|\Omega|^2 - |S|^2\right] \ > \ 0 \tag{2.3}$$

It is seen that this definition essentially defines a vortex as that portion of the flow where the rate-of-rotation is higher than the rate-of-strain. It is a useful feature as it can help to isolate the large vorticies in the flow without capturing the vorticity in the boundary layer where rotation rate and strain rate are of similar magnitudes.

Viscous effects do not account for all sources of vorticity in the flow, however. Inviscid flows can still exhibit rotation due to the interference of certain changes in geometry. The most common examples of this being at the wingtip or the strakes. The flow on the bottom of a lift-producing wing (and similarly a strake) has higher pressure than the air on the top of the wing. At the wingtips, however, high-pressure air from the bottom of the wing sneaks around the tip to the top of the wing as seen in Figure 2.4. The resulting vorticity that is produced can be seen in Figure 2.5.

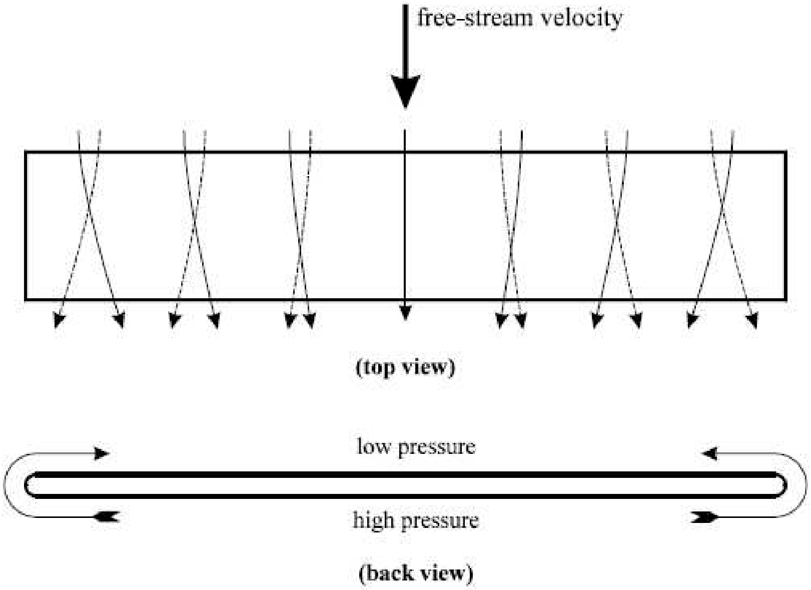


Figure 2.4: Airflow from bottom to top of wing [42].

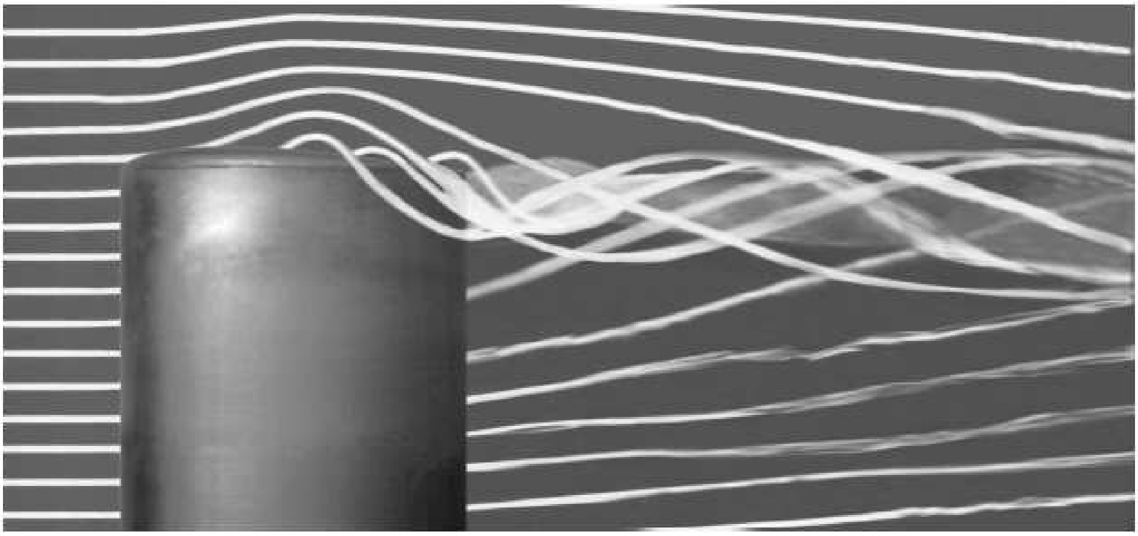


Figure 2.5: Vorticity produced at tip of wing [42].

2.1.4 Boundary Layer Theory. Before a discussion in turbulence theory begins, it is pertinent to discuss turbulent boundary layer theory. Most of the relationships for turbulent boundary layers are primarily based on empirical correlations of the shape of the velocity profile near the wall. This velocity profile is only relevant for viscous flows where a no-slip boundary condition is enforced - zero-velocity at the wall as discussed in Section 2.1.3. An important note is that the boundary layer is initially laminar. This laminar profile is organized, predictable, and stable. In this laminar flow, the linear viscous forces of the fluid dominate the nonlinear inertial forces due to fluid velocity. This is considered low Reynolds number flow where the Reynolds number is the ratio of inertial to viscous forces as seen in Eq. (2.4) [41]

$$Re_x = \frac{\rho U_\infty x}{\mu} \tag{2.4}$$

where ρ is the density, U_∞ is the freestream velocity, x is the characteristic length from the edge of the geometry, and μ is the dynamic viscosity of the fluid. Around Reynolds numbers of 500,000, the flow transitions from laminar to turbulent. Most aerospace applications have Reynolds numbers much higher than that for most of the flow field so most CFD solvers assume fully turbulent flow for the entire domain. For this purpose, only turbulent flow theory will be presented here.

Turbulent boundary layers are generally described in terms of non-dimensional parameters. The first parameter to define is the thickness of the boundary layer, δ (Eq. (2.5)). Describing this equation in words, this is defined as the height off the surface where the flow velocity is 99% of the freestream. In all of the following, the subscript e stands for this "edge" of the boundary layer and w stands for a value at the wall or surface [41].

$$\frac{\delta}{x} = \frac{0.371}{Re_x^{1/5}} \tag{2.5}$$

Similarly, there is an integral parameter across the velocity profile known as displacement thickness, δ^*, akin to boundary layer thickness:

$$\delta^* = \int_0^\infty \left(1 - \frac{\rho u}{\rho_e u_e}\right) dy \tag{2.6}$$

It is also useful to define functions that describe (in non-dimensional form) the velocity at a given point above the surface, u (non-dimensional value is u^+), and the height above the surface, y (non-dimensional form is y^+). These are given as Eq. (2.7) and Eq. (2.8) respectively [41].

$$u^+ = \frac{u}{u_\tau} \tag{2.7}$$

$$y^+ = \frac{\rho_w u_\tau y}{\mu_w} \tag{2.8}$$

Where u_τ is the friction velocity:

$$u_\tau = \sqrt{\frac{\tau_w}{\rho_w}} \tag{2.9}$$

In Eq. (2.9), τ_w is the wall shear stress defined by:

$$\tau_w = \mu \left.\frac{\partial u}{\partial y}\right|_w \tag{2.10}$$

The above equations do not provide a method for calculating the velocity as a function of height off the surface as Eq. (2.7) is only the means of non-dimensionalizing u. Before those equations are provided, it is important to understand that the change in u^+ varies depending on the height off the surface and not just one equation can accurately define the value. Because of this, the boundary layer velocity profile is divided into subregions. They are the sublayer, buffer layer, log layer, and the wake. The first three are defined to be in the “Inner region” and the wake is said to be in the “Outer region” as seen in Figure 2.6.

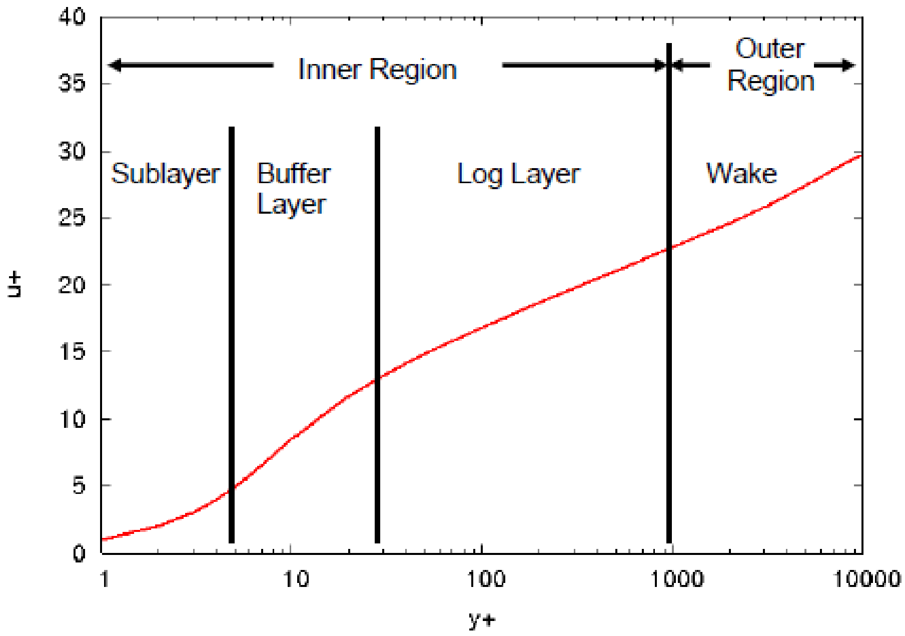


Figure 2.6: Boundary layer regions [41]

The smallest to largest inner region layers are defined respectively by the following:

$$0 < y^+ < 5 \qquad u^+ = y^+ \tag{2.11}$$

$$5 < y^+ < 30 \qquad u^+ = 5 \ln y^+ - 3.05 \tag{2.12}$$

$$30 < y^+ < 1000 \qquad u^+ = \frac{1}{\kappa} \ln y^+ + B \tag{2.13}$$

There is debate over what the values of κ and B should be in Eq. (2.13), but they are often accepted to be 0.4 and 5.5 respectively [41].

The last region to define the velocity profile for is the wake. The purpose in distinguishing this as the outer region as compared to the other three regions is that the inner regions are independent of the large freestream properties “far” from the surface. In comparison, the outer region is much more dependent on the freestream

Reynolds number and pressure gradient.

$$u^+ = \frac{1}{\kappa} \ln y^+ + B + \frac{2\Pi}{\kappa} \sin^2\left(\frac{\pi}{2}\frac{y}{\delta}\right) \tag{2.14}$$

Where Π is given by:

$$\Pi = 0.8(\beta + 0.5)^{0.75} \tag{2.15}$$

$$\beta = \frac{\delta^*}{\tau_w}\frac{\partial p_e}{\partial x} \tag{2.16}$$

This brief overview of turbulent boundary layer theory aids in the understanding of turbulence theory (Section 2.1.5) as well as turbulence modeling (Section 2.1.6) [41].

2.1.5 Turbulence Theory. Turbulence is a three-dimensional, unsteady, non-linear, viscous phenomenon that occurs in high Reynolds number flows. Turbulent flows can be thought of as a series of eddies of widely differing sizes which constantly interact with each other. An example of these turbulent structures on a flat plate can be seen in Figure 2.7. These structures are created by the viscous interaction of the fluid and the wall, and the largest eddies are produced from a conversion of the mean flow turbulent kinetic energy (k). This energy is then transfered to smaller and smaller scales (called the energy cascade) until the scales are so small that the viscosity at the wall dominates and converts the kinetic energy into heat. Figure 2.8 shows a generic depiction of the energy production and dissipation within these flows (note the wave number is inversely proportional to the turbulent length scales [41]).

These differing length scales can correlate to the various regions in the boundary layer as discussed in Section 2.1.4 (notice Figure 2.6 has y^+ plotted on a log scale on the x-axis). In order to describe turbulent flow it is necessary to first quantify the ranges of these length (and corresponding time) scales. The outer scales are determined by the flow velocity and geometry of the problem whereas the inner length scales are set by Reynolds number and the location of where the kinetic energy is balanced by

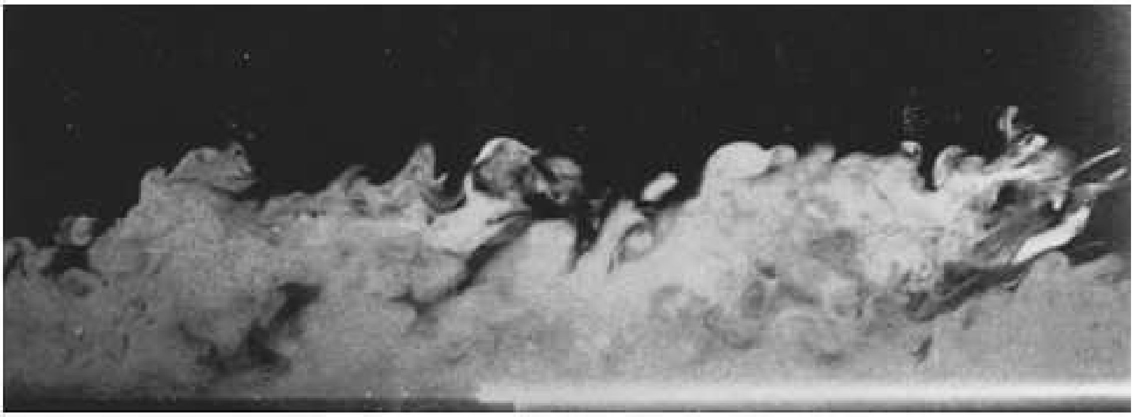


Figure 2.7: View of turbulent structures on plate [41].

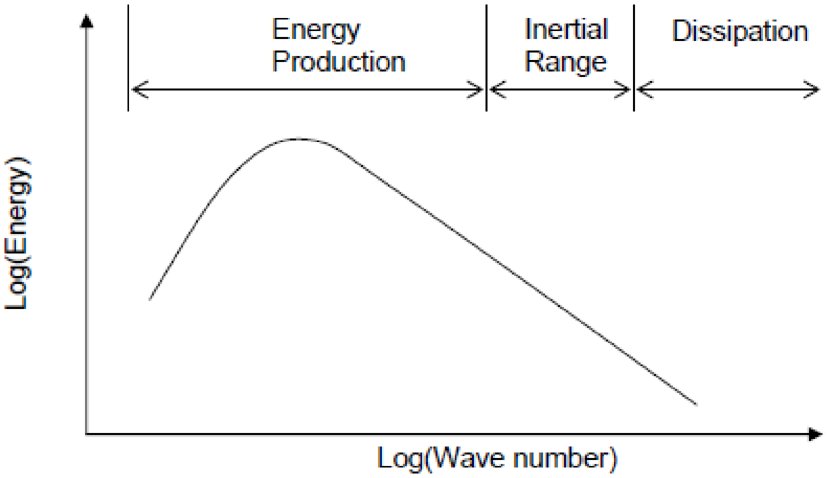


Figure 2.8: Energy production and dissipation in turbulent flow [41].

viscous dissipation, ϵ. The large (L) and small (λ) length scales are found in Eq. (2.17) and Eq. (2.18) respectively [41].

$$L = \frac{k^{\frac{3}{2}}}{\epsilon} \tag{2.17}$$

$$\lambda = \left(\frac{\nu^3}{\epsilon}\right)^{\frac{1}{4}} \tag{2.18}$$

$$\frac{L}{\lambda} = Re^{(}3/4) \tag{2.19}$$

Where ν is the kinematic viscosity, and Re is the turbulent Reynolds number. The length scales correspond to related large (T) and small (τ) time scales which are respectively:

$$T = \frac{k}{\epsilon} \tag{2.20}$$

$$\tau = \left(\frac{\nu}{\epsilon}\right)^{\frac{1}{2}} \tag{2.21}$$

It is seen from the above equations that the range in length and time scales can vary by orders of magnitude for high Reynolds number flows (in which k is much larger than ν). These length scales are also much smaller than the lengths found on an aircraft (such as the length of the wing or fuselage). This will also be an important factor for grid generation as discussed in Section 2.2.1.

It is now important to discuss the governing equations for fluid flow - the Navier-Stokes (N-S) equations. The derivation of these equations is not necessary here and it will suffice to say that Eq. (2.22) is a compact form of three separately applied conservation laws - mass, momentum, and energy - for a fixed control volume [8].

$$\frac{\partial}{\partial t}\int_{\Omega} \vec{W} d\Omega + \oint_{\partial\Omega} (\vec{F}_c - \vec{F}_v) dS = \int_{\Omega} \vec{Q} d\Omega \tag{2.22}$$

Where $\vec{W}$ is the vector of conserved variables (Eq. (2.23)), $\vec{F}_c$ is the vector of convective fluxes (flow through the surfaces of the control volume Eq. (2.24)), $\vec{F}_v$ is the vector of viscous fluxes (Eq. (2.26)), and $\vec{Q}$ contains all of the source terms (Eq. (2.30)). Also, Ω denotes a control volume and dS the surface of the control volume.

$$\vec{W} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E \end{bmatrix} \tag{2.23}$$

$$\vec{F}_c = \begin{bmatrix} \rho V \\ \rho u V + n_x p \\ \rho v V + n_y p \\ \rho w V + n_z p \\ \rho H V \end{bmatrix} \tag{2.24}$$

Where V is the contravariant velocity (the velocity normal to the surface element) and is defined by:

$$V = \vec{v} \cdot \vec{n} = n_x u + n_y v + n_z w \tag{2.25}$$

$$\vec{F}_v = \begin{bmatrix} 0 \\ n_x \tau_{xx} + n_y \tau_{xy} + n_z \tau_{xz} \\ n_x \tau_{yx} + n_y \tau_{yy} + n_z \tau_{yz} \\ n_x \tau_{zx} + n_y \tau_{zy} + n_z \tau_{zz} \\ n_x \Theta_x + n_y \Theta_y + n_z \Theta_z \end{bmatrix} \tag{2.26}$$

With

$$\Theta_x = u\tau_{xx} + v\tau_{xy} + w\tau_{xz} + k\frac{\partial T}{\partial x} \tag{2.27}$$

$$\Theta_y = u\tau_{yx} + v\tau_{yy} + w\tau_{yz} + k\frac{\partial T}{\partial y} \tag{2.28}$$

$$\Theta_z = u\tau_{zx} + v\tau_{zy} + w\tau_{zz} + k\frac{\partial T}{\partial z} \tag{2.29}$$

Which denote the viscous stresses and the heat of conduction.

$$\vec{Q} = \begin{bmatrix} 0 \\ \rho f_{e,x} \\ \rho f_{e,y} \\ \rho f_{e,z} \\ \rho \vec{f_e} \cdot \vec{v} + \dot{q}_h \end{bmatrix} \tag{2.30}$$

In the above: ρ denotes the density; u, v, and w denote the velocity in the x, y, and z coordinate directions respectively; E is the total specific energy; H is the total enthalpy; n_i terms are the components of the unit normal vector of the face in the x, y, and z directions; τ_{ij} terms are the components of the viscous stress tensor; T is the temperature; k is the thermal conductivity, $f_{e,i}$ terms are the components of the external volume forces; and $\dot{q}_h$ is the heat flux due to radiation, chemical reactions, etc.

The above N-S equations for Newtonian fluid flow are what must be solved within all of the length scales and with a time step small enough to capture all of the time scales for a time-accurate solution to fluid flow around any object. As will be discussed in Section 2.1.6, this can be infeasible, so often these equations are simplified by a process known as Reynolds Averaging. This is a process of decomposing the flow velocities and pressure into mean and fluctuating components:

$$v_i = \overline{v}_i + v_i^{'}, \qquad p = \overline{p} + p^{'} \tag{2.31}$$

Where the mean values have the bars and fluctuating values have the primes. These equations are then substituted into the N-S equations which are then solved for the mean values which are often the most interesting for engineering applications [8]. This set of equations are known as the Reynolds Averaged Navier-Stokes (RANS) equations which produces the solution for the mean flow only. These substitutions and simplifications will not be performed here but can be found in Nichols [41]. The bottom line from this process is that the N-S equations are essentially unchanged aside from the of one term known as the Reynolds-stress tensor [8]:

$$\tau_{ij}^{R} = -\rho\overline{v_i'v_j'} = -\begin{bmatrix} \rho\overline{(u')^2} & \rho\overline{(u'v')} & \rho\overline{(u'w')} \\ \rho\overline{(v'u')} & \rho\overline{(v')^2} & \rho\overline{(v'w')} \\ \rho\overline{(w'u')} & \rho\overline{(w'v')} & \rho\overline{(w')^2} \end{bmatrix} \tag{2.32}$$

This tensor represents the transfer of momentum due to the fluctuations of the turbulence. Also, this is a symmetrical matrix given in Eq. (2.32) which makes for only six instead of nine independent components. The addition of this term into the N-S equations therefore requires the introduction of six additional relations to close them. This is the fundamental problem of modeling turbulence for the RANS equations.

2.1.6 Turbulence Modeling. The purpose of this section is not to give a detailed explanation of various turbulence models, but rather to give a brief highlight of the options available to aide in the decision of a final turbulence model.

It was mentioned in Section 2.1.5 that the full N-S equations can be used to fully describe turbulent flows [41], but that in order to do so, all length (see Eq. (2.17) & Eq. (2.18)) and time (see Eq. (2.20) & Eq. (2.21)) scales must be accurately captured. There is a CFD method of accurately capturing all these scales and is known as Direct Numerical Simulation (DNS). DNS CFD simulations are therefore, not modeling turbulence at all, but rather directly solving the N-S equations on a grid with spacing at least as small as the smallest length scales and a time step at least as small as the smallest time scale. It is easy to see how this can quickly increase computational cost as Reynolds number and size of domain increase [41] [47].

Section 2.1.5 posed a common method of a time-averaging the N-S equations in order to bypass the necessity of capturing all of the length and time scales. This method produces the RANS equations and that section left off by displaying the six additional Reynolds stress tensor terms that need to be modeled. These types of turbulence models that rely on this method are simply known as RANS turbulence models. They are based off of one single length scale and require various modeling techniques to fulfill the closure problem [41].

As stated in the beginning of this section, it is not the goal to present detail about the various turbulence models, but fundamental to so many RANS turbulence models is the approximation made by Boussinesq [8] [41]. He proposed that the turbulent stresses can be treated similarly as the viscous stresses in laminar flow. That is - the Reynolds stresses can be approximated as being proportional to the local mean flow rate-of-strain (Eq. (2.1)):

$$-\rho\overline{v_i'v_j'} = 2\mu_t S_{ij} - \frac{2}{3}\rho k\delta_{ij} \tag{2.33}$$

Where δ_{ij} is the Kronecker delta function. Boussinesq's approximation reduces the number of required models from the six needed for the Reynolds shear stress tensor, to one [41]. This one value is deemed the eddy viscosity, μ_t, even though it is not actually a characteristic of the fluid at all. It is instead a function of the local flow conditions [8].

There have been multiple turbulence models based off the Boussinesq approximation. Some of these use algebraic equations to model the eddy viscosity but are generally quite limited in their use. Some of the most common, however, employ one or two transport equations to model various properties upon which the eddy viscosity is solved [41] [8] [47]. The general form of a transport equation is:

$$\frac{\partial Z}{\partial t} + U_i\frac{\partial Z}{\partial x_i} = \frac{1}{\sigma}\frac{\partial}{\partial x_j}\left[(v + v_t)\frac{\partial Z}{\partial x_i}\right] + P(Z) - D(Z) \tag{2.34}$$

Where Z is the variable being modeled, the left-hand side is the convective transport of Z, the first term in the right-hand side is the diffusion term, $P(Z)$ is the production term, and $D(Z)$ is the destruction term [41].

The most popular one-equation model for external flow is the Spalart-Allmaras model. It was derived by using empirical relationships, Galilean invariance, and dimensional analysis, and was calibrated from results to two-dimensional mixing layers, wakes, and flat-plate boundary layers [41] [8]. The Spalart-Allmaras model utilizes the transport equation to model a kinematic eddy viscosity parameter from which the eddy viscosity is estimated. This model is easy to implement into any type of grid, it is robust, converges fast to steady-state, and only requires moderate grid resolution in the near-wall region [8]. This model, however, displays shortfalls in modeling turbulence in separated flow regimes and even limitations in some time-accurate cases [41].

The next turbulence model does not rely on the RANS equations, but instead is three-dimensional, time-dependent solution to the N-S equations, similar to DNS. The primary difference between this turbulence model and DNS is that it takes advantage of the universal character of the small turbulent structures (as mentioned when discussing the inner-region in Section 2.1.4). This model is known as the Large-Eddy Simulation (LES) model as it still captures the large length scales (eddy's) of turbulent flow to calculate the momentum and energy transfer from the energy-carrying structures. Since LES does not require the capture of the small length scales, it provides a significant computational saving over DNS but even still, when considering high Reynolds number flows at transonic speeds, this method is infeasible [8]. LES provides high-fidelity results, though, even in highly separates flows [45]. Variants of the original LES model include the use of wall functions to save on even more computational expense and other even utilize a hybrid RANS/LES model.

Probably the most common hybrid RANS/LES model is the Detached Eddy Simulation (DES) model. The DES model provides a significant computational savings by greatly reducing the number of cells required, which makes high-Reynolds number applications possible to run [45]. The benefits are quite obvious in that DES can still capture the large unsteady eddies in separated flow in a time-accurate three-dimensional solution while cutting cost by only employing the LES solver out of the boundary layer. There are pitfalls, still, in that it does still require a very fine grid with very specific requirements for what will produce the best results as discussed in the Young Person's Guide by Spalart [45]. Also, there are known issues to DES with "ambiguous grids." That is to say, there are issues when the defining line between the RANS region and LES region are ambiguous in the domain [45]. Said another way, DES performs well in flows where there are either thin boundary layers or in regions of massive separation, but in thick boundary layer regions or "shallow" separation, problems are known to arise [46].

In order to help fix this issue, a new version of the DES turbulence model was created in which the switch from RANS to LES is modified to change at the height of the boundary layer so the RANS model is kept through this entire region. For the thick boundary layer regions this results in a delay for the switch from RANS to LES and because of such this new model is known as the Delayed DES (DDES) turbulence model [46].

2.2 CFD Settings

The underlying premise of this thesis is that CFD is a proven method of providing high-fidelity aerodynamic data for a full, high-performance aircraft - even when operating in nonlinear regions. A large body of research has been performed by researchers at the US Air Force Academy and the US Air Force SEEK EAGLE Office in which high-fidelity results are obtained from CFD [11, 16–19, 35, 39]. Within these papers, the unstructured mesh solver, Cobalt (see Section 2.2.2), is used coupled with the Detached-Eddy Simulation turbulence model (see Section 2.1.6), and adaptive mesh refinement (see Section 2.2.1).

The purpose of this section is to outline the methodology to be used in performing CFD calculations of a high-fidelity nature. Some of the key components to discuss are the quality of the grid, the accuracy of a chosen turbulence model, and the use of an appropriate solver and it settings. These three main topics are interrelated and the choices of which are dependent on the expected flow features as well as the object to be modeled.

2.2.1 Grid Development. Grid generation is often the most time consuming element in the CFD process, at least when modeling an entire aircraft. The complete process of initial grid generation is beyond the scope of this research, but the grid itself (especially specific aspects of it), can have a great influence on the results gained from CFD.

A common approach to the grid development process is to start with an initial grid which accurately describes the boundary surface of the object to be modeled and to follow certain rules of thumb when creating the volume grid [13]. For instance, some interrelated rules of thumb include having the first grid point located away from the wall less than a $y^+ = 1$ value, have at least two or three grid points in the viscous sublayer for accurate shear prediction, and to have a growth rate of the boundary layer cells of approximately 1.25 or less [13]. It is then common to use various techniques to cluster cells in expected wake regions to the large separation can be modeled. Once the initial grid is complete, a solution is found, the grid is refined (often by splitting each edge in the domain to refine everywhere or some other type of global refinement) and the solution run again. This process is done successively until all the appropriate flow features emerge and the final solution changes less than a desired amount [13].

The above process can be very computationally expensive as the grid cell count increases drastically at each iteration. Even still, Cummings, et al. [13] propose that the above process may not actually produce more accurate results and instead a physical basis for choosing the appropriate grid should be employed for accurate flow prediction. One proposed idea is to produce the initial grid essentially the same as mentioned above but then to refine the grid using a method called Automatic Mesh Refinement (AMR). This method utilizes the flow solution of the initial grid to define regions of highest separation and rotation by tracking vorticity (for example). The cells within a defined iso-surface of vorticity are then removed and then re-grown at a scale factor of 0.5 [13]. Both of the above processes were used in [13] and Figure 2.9 shows a comparison between the most refined grid (with 10.7 million cells) and a grid created with two levels of AMR based off of an initial grid with roughly 2 million cells and resulted in a grid with 3.2 million cells. This figure show how the vortex

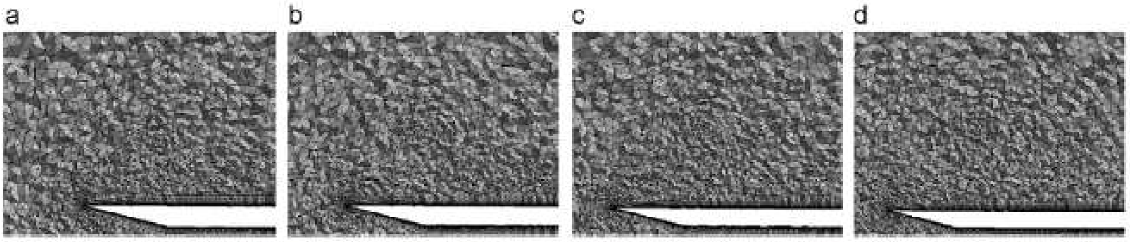


Fig. 13. Crossplanes of Grid 4 (10.7 million cells) at four chordwise stations [19]: (a) $x = 400$ mm; (b) $x = 500$mm; (c) $x = 600$ mm and (d) $x = 700$ mm.

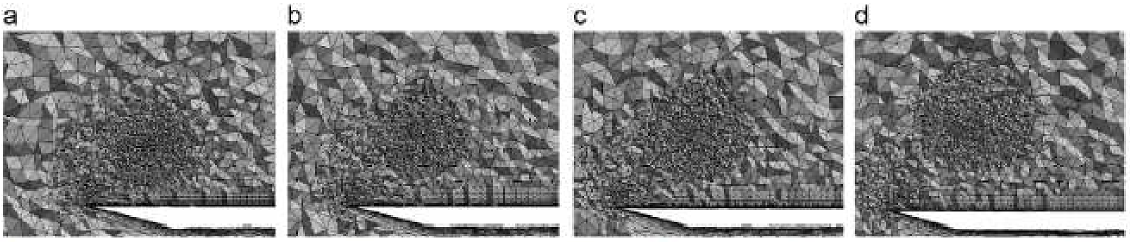


Fig. 14. Crossplanes of Grid 5 (3.2 million cells) at four chordwise stations [19]: (a) $x = 400$ mm; (b) $x = 500$mm; (c) $x = 600$mm and (d) $x = 700$ mm.

Figure 2.9: Comparison of very fine grid (top) to grid created with AMR (bottom) [13]

core is even more refined in the AMR grid than it is in the very fine grid. Figure 2.10 shows the results of the solution of these grids (very fine bottom left and AMR bottom right) and the other less refined iterations of the very fine mesh (top). The AMR grid shows more coherent vortical structures relative to even the very fine grid. There is also a large amount of three-dimensional structure in the core region post vortex breakdown. Cummings, et al. determined for this high AoA (27 deg) delta wing the AMR grid produced equivalent results to the very fine grid [13]. The same paper [13] also discusses the use of AMR on an F-18C grid with accurate results when used in tandem with a DES turbulence model (see Section 2.1.6).

2.2.2 Flow Solver and Settings. Kestrel is an Air Force developed finite volume, cell-centered, unstructured mesh solver based on the Air Vehicles Unstructured Solver (AVUS) developed by AFRL/RBAC in the 1990s (Also, the AVUS solver is comparable to many ways to the industry code, Cobalt, which was mentioned in Section 2.2). See page 51 in the Kestrel User Guide (Version 2.0) for more details

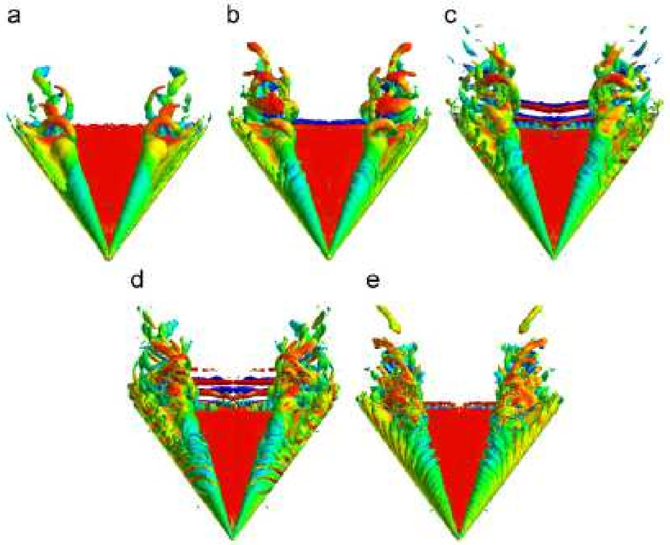


Fig. 16. Iso-surfaces of vorticity magnitude colored by spanwise vorticity; $\alpha = 27^{\circ}$, $Re_c = 1.5 \times 10^6$ [19]: (a) grid 1; (b) grid 2; (c) grid 3; (d) grid 4 and (e) grid 5.

Figure 2.10: Comparison of results of successively refined grids (top), fine grid (bottom left), and AMR grid (bottom right) [13].

about the solver such as the method of solving the exact Riemann Solver, obtaining second order accuracy, types of temporal schemes, etc [12].

The Spalart-Allmaras one-equation turbulence model, the Wilcox $k - \omega$ and Menter's Baseline two-equation models, and the DDES/Spalart-Allmaras turbulence models are all implemented in the Kestrel solver (known as kAVUS). The AVUS solver, however, had multiple functions built-in, whereas Kestrel was developed to be a very general architecture that can handle new components and more complicated use cases which can be developed and added at later dates. Because of this, the kAVUS solver has taken out features built into AVUS such as reading and splitting the grid and force and moment calculation so that the only purpose of kAVUS is to compute the solution at a single iteration [12].

One important feature of Kestrel is the graphical user interface from which the user inputs the solver settings. From this, the range of inputs and use cases can be decided. The ability to select between static or dynamic simulations (6DOF or prescribed motion) is available. For this research, the prescribed motion features will be utilized for inputing the designed maneuvers for the solver to compute.

2.3 Stability Derivatives

In Chapter I, the importance of accurately predicting aerodynamic data and stability derivatives in nonlinear regions was established. The purpose of this section is to outline what parameters are to be gathered, what they are, and what they mean about the aircraft. For longitudinal motion (which is the only motion for this research), there are two aerodynamic forces and one moment which act on the aircraft. They are lift (the forces acting perpendicular to the freestream flow), drag (the forces acting parallel to the freestream flow), and pitching moment (the resulting moment in the lift-drag plane after all longitudinal forces on the aircraft are summed). It is common to non-dimensionalize these values, however, to express them as coefficients. The lift (C_L), drag (C_D), and pitching moment (C_m) coefficients are displayed in the following:

$$C_L = \frac{L}{\frac{1}{2}\rho_\infty V_\infty^2 S} \tag{2.35}$$

$$C_D = \frac{D}{\frac{1}{2}\rho_\infty V_\infty^2 S} \tag{2.36}$$

$$C_m = \frac{m}{\frac{1}{2}\rho_\infty V_\infty^2 S l_{ref}} \tag{2.37}$$

Where L is the total lift force, D is the total drag force, m is the total pitching moment, ρ_∞ is the freestream fluid density, V_∞ is the freestream velocity, S is a reference area (generally the planform area of the aircraft), and l_{ref} is a longitudinal reference length (often the mean aerodynamic chord) [42].

The lift on an airplane is of obvious importance - it gets and keeps the airplane in the air. It is also the force used to control and maneuver the airplane along with thrust. It is through controlling and changing the direction and magnitude of these forces that this control and maneuvering is accomplished [42]. Drag also plays an important role in this factor as it and weight make up what can be deemed "resistive" forces to thrust and lift and an accurate knowledge of them is also essential in the design of an airplane. This knowledge factors into the design of the flight control system (which exists on essentially all high-performance aircraft), determining the controllability and maneuverability of the aircraft, structural design, and many other areas.

The pitching moment on the aircraft also plays into many of the above areas, especially control system design. Longitudinal stability is primarily determined by the pitching moment. Static stability can be explained as a restoring to equilibrium of a system after a perturbation is input. If the system departs for equilibrium following a disturbance, then it is said to be negatively stable and if it remains at the perturbed stated then it is neutrally stable. All three of these states are depicted in Figure 2.11. In terms of pitching moment values, a quick examination of aircraft response can explain desired values. Consider a perturbation in which the AoA of an aircraft

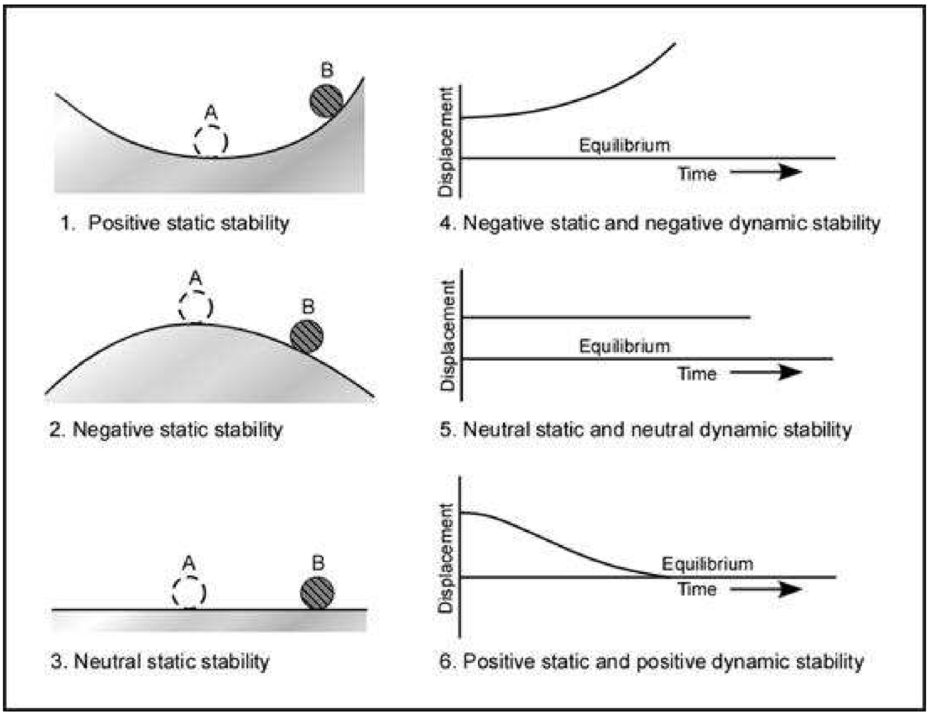

A
B
1. Positive static stability
Displacement
Equilibrium
Time
4. Negative static and negative dynamic stability
A
B
2. Negative static stability
Displacement
Equilibrium
Time
5. Neutral static and neutral dynamic stability
A
B
3. Neutral static stability
Displacement
Equilibrium
Time
6. Positive static and positive dynamic stability

Figure 2.11: Depiction of Static stability, neutrality, and instability [2]

is increased from a trim scenario. A stable system will have characteristics in which the AoA will decrease “on its own.” In other words the pitching moment must be negative with an increase in AoA so as to “push” the AoA back down. Likewise, if the AoA is decreased then the pitching moment must be positive in order to increase the AoA back to equilibrium. A convenient way to determine static longitudinal stability, then, is to examine the slope of the pitching moment with AoA (know as $C_{m,\alpha}$) [42]. This is depicted in Figure 2.12.

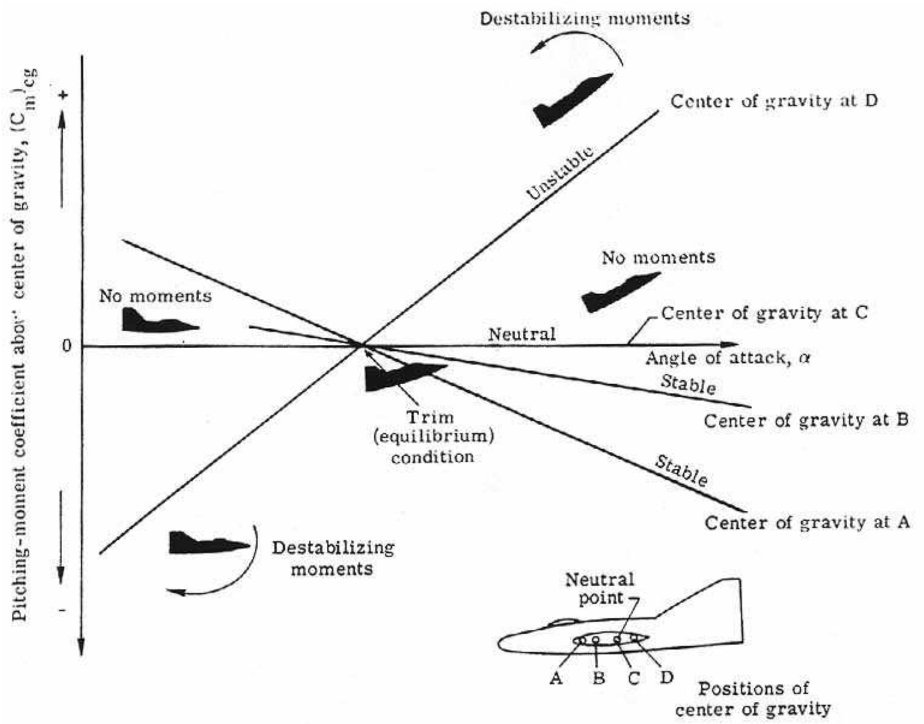


Figure 2.12: Slope of pitching moment for static stability, neutrality, and instability [1]

It may seem odd that only the lift, drag, and pitching moment coefficients will be modeled then. This is due to the fact that other desired derivatives (such as $C_{m,\alpha}$) can be obtained by simply taking the derivative of the model corresponding to the desired result (for example: take the derivative of the pitching moment coefficient model with respect to AoA to obtain a model for $C_{m,\alpha}$). In this way, models of many different stability coefficients can be obtained for “free” from the initial models.

The discussion of stability would not be complete without at least mentioning dynamic stability as well. Where static stability is a returning to an equilibrium state, dynamic stability addresses the motion associated with this return (or departure) to equilibrium. A system may tend to return to the equilibrium state, but does it overshoot this destination? If so, is the magnitude of the overshoot larger than the initial pertibation? If so then the system is dynamically *un*stable. If not then it is dynamically stable. Figure 2.13 depicts these dynamic states of equilibrium.

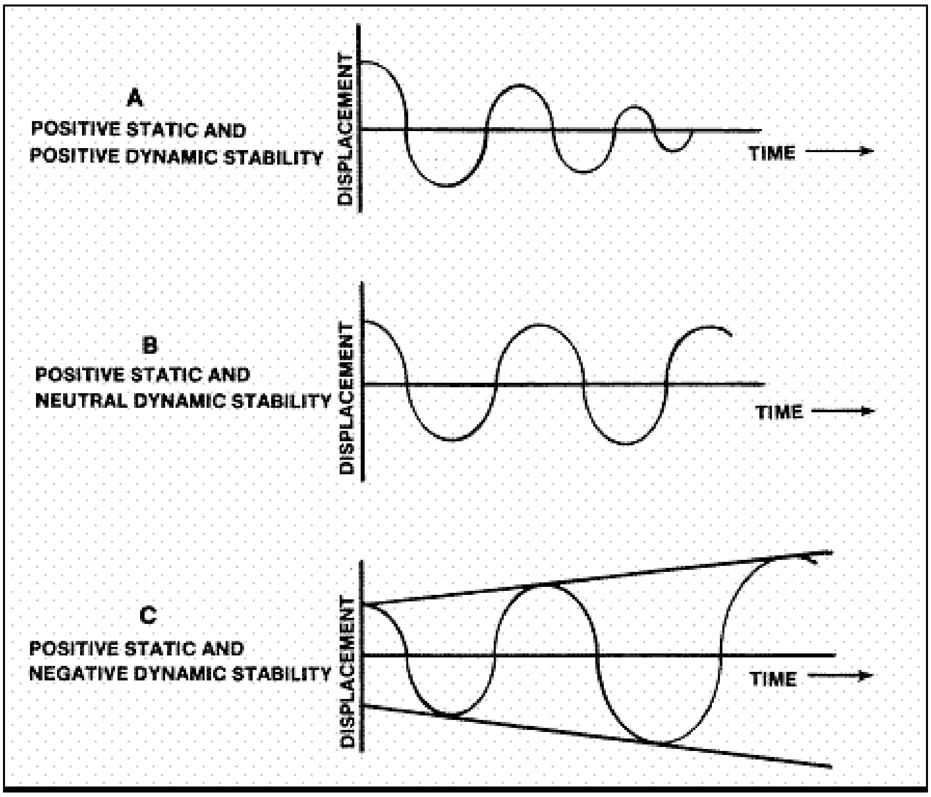


Figure 2.13: Depiction of dynamic stability, neutrality, and instability [2]

The natural stability of the aircraft at any expected orientation is also of great importance when designing a control system, especially for high-performance aircraft. They are often naturally unstable to allow for high maneuverability as stability and maneuverability are in conflict with each other [42] [29].

As a last note, although only the longitudinal forces and moment were mentioned in this section, the same ideas and principles apply similarly when considering longitudinal and lateral flight dynamics together.

2.4 System Identification

Section 1.2.4 discussed some of the various methods for modeling aerodynamic data with the result that System Identification is the best method for understanding the actual dynamics associated with the aircraft [21]. SID is the process of creating a mathematical model of a system based on the responses of the system to various inputs. The structure of this mathematical model can vary widely to accommodate the type of system (linear versus nonlinear, etc) and its intended use. Figure 2.14 shows a simplified diagram of the task of SID to model an unknown, nonlinear aircraft system based on its inputs and outputs. Fortunately, much work has been done in the area of SID for nonlinear aircraft such as [23,24,26,29–31,33]. From these results, there are a variety of forms that can be expected for the system model to take and as well as there can be some advanced insight into which input parameters and coupled parameters tend to represent specific output parameters the best [33].

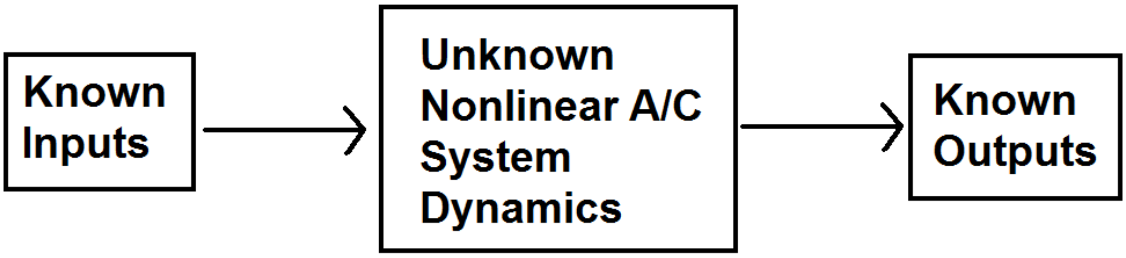


Figure 2.14: Block diagram of relationship of system to known inputs and outputs.

CFD simulations are well suited to SID modeling as the input and output states are all measured during the duration of the maneuver, there is very tight control of the input variables, and CFD does not rely on sensors to gather output data. Quite often the entire output state is not directly measured and must be estimated based on correlations between the known inputs/outputs and whatever is known about the system. This state estimation can introduce error especially as the sensor data of the known inputs/outputs may have a significant amount of error themselves [33]. From this error, the flow conditions are not exactly known whereas in CFD they are tightly controlled and the overall level of noise is reduced.

2.4.1 SID Techniques. One of the most common methods for SID is the global nonlinear parameter modeling technique which describes the dependence between the motion (input variables) and the computed aerodynamic forces and moments (the output). The goal being to find the model which accurately captures the nonlinearities with the minimum number of terms. This latter requirement improves the ability to identify the model parameters which will result in the best predictive capability [15, 21].

The multivariate analysis technique uses orthogonal modeling functions generated from the data to determine a nonlinear structure for the model and expands the retained modeling functions into a multivariate polynomial [21, 30]. One of the main advantages of this technique is that the dependencies of the output variables on the input variables is done automatically based on data and statistical modeling metrics without the need for a user's judgment [21].

Another technique for SID is stepwise regression. It is also an automated procedure for statistical model selection, but for cases where there are a large number of potential regressors (input variables). In this technique, the set of input variables which produce the best fit according to some metric (such as the R^2, or coefficient of determination, statistic as seen in Eq. (2.38)) is found.

$$R^2 = \frac{S_t - S_r}{S_t} \tag{2.38}$$

where

$$S_t = \sum (y_i - \overline{y})^2 \tag{2.39}$$

$$S_r = \sum (y_p - y_i)^2 \tag{2.40}$$

With y_i being the known "true" value, $\overline{y}$ the average of the true values, and y_p being the predicted value from the model at that point.

2.4.2 System Identification Program for Aircraft. One specific collection of SID tools that utilizes the techniques described above is called System Identification Program for Aircraft (SIDPAC). This specific collection of programs is discussed due to its ability to create accurate models (due to a high-fidelity method employed) and because it is the bases of the SID program implemented directly into Kestrel. This code was developed using `Matlab`® at the NASA Langley Research Center and has been utilized successfully for SID modeling for many different flight test programs and wind tunnel experiments [32]. It includes routines for "experiment design, data conditioning, data compatibility analysis, model structure determination, equation-error and output-error parameter estimation in both the time and frequency domains, real-time and recursive parameter estimation, low order equivalent system identification, estimated parameter error calculation, linear and nonlinear simulation, plotting, and 3-D visualization" [32].

SIDPAC decomposes a signal into a deterministic signal and nondeterministic noise using Fourier analysis and the optimal Weiner filter (a low-pass filter). The cut-off frequency for the low-pass filter is chosen automatically based on the data provided and the filtered data is what is actually modeled in SIDPAC using multivariate orthogonal modeling functions generated from the independent variables. This method is compared against a stepwise regression model in [21] and it is found that the SIDPAC technique of multivariate orthogonal functions produces much better results.

2.5 Previous Research

All of the requisite topics have now been covered relating to the research of in this paper. The last step of this chapter is to examine what work has been accomplished in this area thus far.

2.5.1 Initial Attempts. In September 2003, a group of almost 100 technical professionals from government, industry, and academia gathered together in Hampton, VA for a NASA-sponsored symposium. The purpose of this conference was to discuss Computational Methods for Stability and Control (COMSAC) and to discuss the current capability of CFD to predict quality S&C data as well as what challenges stand in the way of doing so. Many of the issues mentioned in Section 1.1 were introduced, but at that time it was said that although CFD and fields such as aerodynamics and performance have had much discussion and crossover, there was a lack of this type of communication between CFD and S&C professionals. Within the aerospace community, it is commonly accepted that the field of CFD has rapidly matured and that the next big pay-offs could occur in S&C [20], but it is clear from the review paper that there was at that time a lack of organization in the quest - which was the purpose of this conference [20, 21].

At the conference, NASA Ames researchers attempted a "brute force" method to filling an S&C database [10, 40, 44]. It was found that to produce a reasonable database, it would include the running of "30 different angles-of-attack, 20 different Mach numbers, and 5 different side-slip angles, each for a number of different geometry configurations or control surface deflections [10]". An emphasis was placed on automating the process and rely on parallel computing as much as possible to reduce the required time but in order to feasibly conduct such a large volume of work would require primarily Euler simulations with fairly coarse Cartesian grids. It was also posed that a select few hundred of the required solutions could be calculated and then an interpolation procedure could be utilized to fill out the missing runs [21].

A similar approach was to turn many of the runs to lower order solutions with few high-order solutions at various locations to cut computational cost. This is a tempting method but relies on efficient knowledge to know when a low-order solution is acceptable and when it is not. It also assumes that the discrete set of points computed capture all of the necessary nonlinearities when it could clearly be missing important information. Another approach used was to apply a potential flow solver to predict S&C derivatives as well as with a serial Euler code. The potential flow solver provided much shorter execution times but suffered from an inability to predict the frequency dependent behavior of dynamic S&C values [21].

Another method of using B-splines to represent nonlinear aerodynamic functions in flight dynamic models was presented by Bruce and Kellett [9]. The method presented offers a fit to an smooth function with guaranteed continuity. It also allows for a possible reduction in the number of flight tests since large portions of nonlinear functions can be identified from large amplitude maneuvers [21].

2.5.2 Current State-of-the-Art. A large body of work has been accomplished in analyzing many maneuvers (often referred to as training maneuvers) of various types [14, 21, 34, 36–38]. The training signals examined include "continuous [AoA] sweeps, sinusoidal pitching motions, coning motions, oscillatory coning motions, configuration plunge pulses, plunge chirps, pitch chirps, Schroeder plunge motions, yaw chirps, and composite pitch-yaw chirps [15]." Since the goal is to accurately model high-performance aircraft, these maneuvers use large amplitude input signals to excite the range of unsteady, nonlinear dynamics of the aircraft. It was found important to cover the entire range of the independent variables for the nonlinear model identification [21].

One paper [21], in particular, is of increased importance for a couple of reasons. First, the initial grid provided for this research is only a slightly modified version of the F-16C grid in this paper (it is the exact grid from [15]) and looking at results and settings used in this paper will lead to an understanding of what to expect from the grid. Second, this paper details the results to a large number of the maneuvers discussed above.

The first set of simulations was to examine the effect of varied constant pitch rate from -5 to 60 deg AoA at pitch rates varying from 5 to 40 degrees per second. It was noted that at even a 5 degree per second pitch rate included dynamic effects on the lift coefficient curve although the slope of which was the same in the linear region (below 12.5 degree AoA for their unspecified flight condition) regardless of the pitch rate. Beyond this linear range, however, this dynamic lift due to constant pitch rate had a significant effect on the results such as the magnitude and location of max lift coefficient [21].

Another maneuver covered in the same paper is the plunge pulse. This maneuver is essentially one cycle of a sinusoidal change in AoA, where the change in AoA is induced by a pure "up and down" translation instead of a rotation. Other maneuvers were also performed such as a Schroeder Sweep plunge and conventional and oscillatory coning motions. Each of which are compared to other maneuvers such as DC chirp plunges and sinusoidal pitching motions which were not used to create the models [21]. The details of these maneuvers are unimportant other than to note the wide variety of options examined to date. Of all of these (and other) maneuvers analyzed, the chirp signals were determined to be the best candidates for creating the ROMs. This "chirp" signal is a sinusoidal motion with varying frequency and often amplitude [15].

2.5.3 Where This Research Fits In.

The work presented in Section 2.5.2 also states the shortcomings of the maneuvers examined so far. The primary shortcoming is in the prediction of the various static coefficients beyond the linear range of AoA [15]. The training maneuvers used are fully dynamic in nature and as such do not include static data. The models do predict dynamic data in nonlinear regions quite well, however, so the insight gained thus far is still of great worth.

The ultimate desire is to generate efficient yet accurate nonlinear aerodynamic models capable of predicting static and dynamic force and moment stability coefficients, so a new maneuver needs to be developed and analyzed that includes both static and dynamic regions. The knowledge gained in using a high amplitude maneuver, built upon the current "best" maneuver of chirp sinusoidal motion, will be used to create a modified maneuver known as a chopped sinusoidal motion. This motion varies the amplitude and the frequency of sinusoidal AoA motion but at each maximum or minimum amplitude the AoA will be held constant for a period of time to introduce these static regions as will be seen in Section 3.4.

III. Methodology

3.1 Overview

The methodology for this research may be broken into five stages: refining the problem, development of metrics, development of maneuvers, finalizing the solver settings, and refining the grid. It has also been deemed proper to limit the work of this thesis to examine longitudinal aircraft responses only as the lessons learned can be translated into all three axes as part of future work.

3.2 Refining the Problem

In order to begin developing metrics or maneuvers or running CFD simulations, the problem must be sufficiently defined. The regressor space, flight conditions, and desired outputs must each be specified as well as a common ground for each maneuver to be compared against.

Practically, the choice of regressor space is determined primarily by some given requirement of what range of AoA, pitch rate, etc are to be modeled. The choice of regressor space for the purpose of this thesis is quite arbitrary, though, since the purpose of this work is not to derive accurate models of the F-16C, but to determine a new training maneuver with better static and dynamic prediction capabilities. So as long as the defined regressor space meets certain criteria, it will be acceptable. First, it must cover a wide range of linear and nonlinear longitudinal aircraft motion. Next, it must provide enough "room" to allow for the large amplitude maneuvers to lie within it. Lastly, there must also be some practical constraints so that the regressor space falls within a set of reasonable/expected values of the real F-16C. In accordance with this, the regressor space will be set as a range from -10 to 30 degrees AoA, and the pitch rate, Q, will be set between ± 100 degrees per second.

In the introduction it was mentioned that two Mach numbers will be evaluated, one subsonic and the other transonic. Needing to also follow guidelines for proper gridding techniques (Section 2.2.1), for proper initial y^+ values, the flight conditions are set at Mach numbers of 0.5 and 0.9 at an altitude of $10,000$ feet and assuming a standard day. For the given F-16C grid, this produces an initial y^+ value of less than 1 for both Mach numbers. This process is backward from what is normally done. Typically, the flight conditions are known ahead of time and the grid is created to give the proper y^+ values, etc.

Section 2.3 gives proper guidance in the desired outputs that should be modeled for longitudinal quantification of S&C derivatives. These values are lift coefficient, C_L, drag coefficient, C_D, and pitching moment, C_m. There are, of course, more longitudinal dynamic derivatives than these that would be useful to S&C engineers, but the models created for these primary variables can be quickly differentiated to provided needed stability and ultimately control derivatives to aid in their analysis [15].

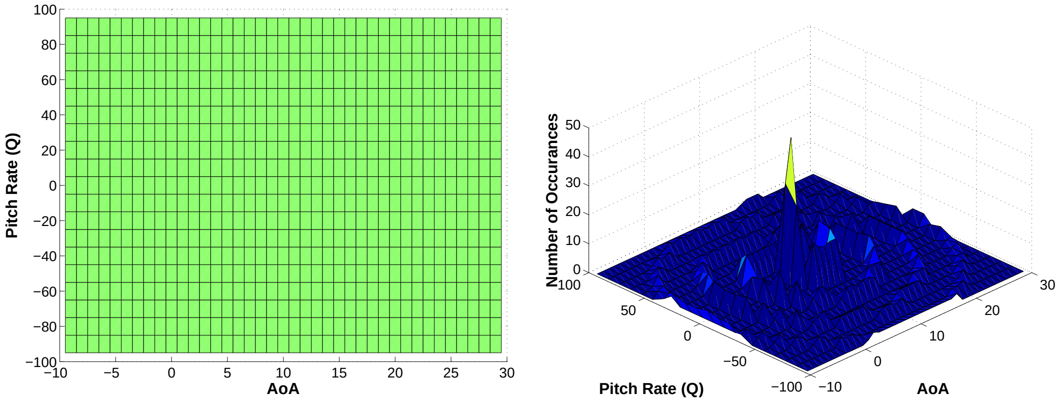
The last item to address in order to properly refine the problem is to establish a common ground for all maneuvers to be compared against. The desire is to find a particular type of maneuver that minimizes the computational time to produce an accurate model. This is essentially an optimization problem where computational expense is the cost function. This type of problem cannot be evaluated by optimization techniques available. Therefore, each maneuver will be set to have the same length in time (a value of five seconds) and no maneuver is allowed to leave the regressor space at any time. In this way, each maneuver has the same real (and equivalently computational) time, and the "best" maneuver will be the one which produces results that fit the comparison data best (from the R^2 technique mentioned in Section 2.4.1).

3.3 Metric Development

Another goal of this thesis is to see if there is a way to determine how good a maneuver is *before* it is run. In an attempt to do so, a series of metrics will be developed to quantify certain aspects of each maneuver to see which are most important (if any certain conclusions can be drawn).

Before a maneuver is run, the only information available is the trace of the maneuver itself. Therefore, these metrics must rely solely on the input regressor values that feed into the system, and if a method can be produced that will predict the "goodness" of a maneuver before it is run, the metrics must be quantifying measures of the path the maneuver traces on the regressor space.

Discretizing the regressor space to a to-be-determined sufficient degree would be the first step in creating these metrics (Figure 3.1(a)). The maneuver input variables could then be plotted on this grid over the regressor space to see how many points are in each cell. An example of this for a simple, constant frequency, linear decrease in amplitude, sinusoidal motion is seen in Figure 3.1(b). These figures show a very coarse grid (step size of 1 deg for AoA and 10 $\frac{deg}{s}$ for pitch rate) for illustration purposes. In reality, the discretization is much finer in order to capture the various regions of the regressor space more accurately, such as values at zero pitch rate. The exact steps in AoA and pitch rate will need to be determined, but not before the metrics are all defined as they may help to define the final step sizes.



(a) Discretized regressor space [3]

(b) Example maneuver plotted [4]

Figure 3.1: Example of the discretized regressor space.

3.3.1 The Metrics. The metrics to be created should attempt to be quantifiable measures of the discretized regressor space. The first metric is then defined as the percentage of the entire regressor space the maneuver covers. This calculation is done by counting the number of cells with any number of points in it at all (whether it has one or many points does not make a difference) and divide this number of cells by the total number of cells as in Eq. (3.1).

$$Metric_1 = \frac{\#\ of\ cells\ with\ data\ points}{Total\ \#\ of\ cells}\ \times\ 100 \tag{3.1}$$

A second metric is defined similarly as the percentage of the cells on the boundary which have any number of data points in them as seen in Eq. (3.2). This metric attempts to quantify how well a maneuver captures the bounds of the regressor space that is being modeled.

$$Metric_2 = \frac{\#\ of\ cells\ on\ boundary\ with\ data\ points}{Total\ \#\ of\ cells\ on\ boundary}\ \times\ 100 \tag{3.2}$$

These first two metrics do not give insight into how evenly the data covers the regressor space, so the next two metrics attempt this. The first looks at the spread in the data across all AoA values. To do this, each column of AoA is checked for the number of cells with any number of values and then divided by the total number of cells in the column. This produces an array of decimal percentage values for each column in the AoA direction. For a perfectly distributed maneuver, each column would have the same number of cells with data in them, producing a standard deviation of zero. That is, then, the final step in calculating the next metric – taking the standard deviation of the percentage of cells with data in them in each AoA column, normalized by the maximum value. To help illustrate this metric, Eq. (3.3) is provided.

$$Metric_3 = std \left[\left(\begin{matrix} \left.\frac{\#\ cells\ w/\ data\ points}{Total\ \#\ cells}\right|_{column_1} \\ \left.\frac{\#\ cells\ w/\ data\ points}{Total\ \#\ cells}\right|_{column_2} \\ \vdots \\ \left.\frac{\#\ cells\ w/\ data\ points}{Total\ \#\ of\ cells}\right|_{column_{last}} \end{matrix} \right)_{AoA\ cells} \div\ max\ value \right] \tag{3.3}$$

Where *std* is the standard deviation function.

The next metric is the same as the previous except it examines the evenness of the data in the rows of pitch rate values. Equation (3.4) illustrates this metric.

$$Metric_4 = std\left[\left(\begin{array}{c} \left.\frac{\#\ cells\ w/\ data\ points}{Total\ \#\ cells}\right|_{row_1} \\ \left.\frac{\#\ cells\ w/\ data\ points}{Total\ \#\ cells}\right|_{row_2} \\ \vdots \\ \left.\frac{\#\ cells\ w/\ data\ points}{Total\ \#\ of\ cells}\right|_{row_{last}} \end{array}\right)_{Q\ cells} \div\ max\ value\right] \tag{3.4}$$

Where Q represents the pitch rate.

The rest of the metrics will now focus on specific regions of interest in the regressor space. The first region of interest is where the maneuver has zero pitch rate. Section 2.5.3 discussed the lack of predictability of static data in the nonlinear regions and the primary focus of this research is to focus on implementing static regions in the maneuvers. Three metrics can be used to help quantify how well this region of the regressor space is covered. First, the overall percentage of the zero pitch rate AoA cells with data in them (Eq. (3.5)).

$$Metric_5 = \frac{\#\ of\ cells\ at\ 0\ Q\ with\ data\ points}{Total\ \#\ of\ cells\ at\ 0\ Q} \times\ 100 \tag{3.5}$$

Next, the average total number of points per cell in the same row of cells (Eq. (3.6)). Note this metric takes into account how many points are in each cell and not whether or not there are any points at all in a cell. In this way, more "credit" is given for extra time spent at zero pitch rate for a given AoA to allow for time to converge to a proper solution. This metric (and future similar metrics) must be taken with caution as spending an extraordinary amount of time at one static AoA value can inflate this number without providing more useful information. Experience must be gained to determine how many iterations or how much physical time is needed to arrive at a converged solution after zeroing pitch rate.

$$Metric_6 = \frac{Total\ \#\ of\ data\ points\ at\ 0\ Q}{Total\ \#\ of\ cells\ at\ 0\ Q} \tag{3.6}$$

The last of the three metrics for zero pitch rate is the standard deviation of the data points in the last metric. This metric normalizes the number of points in each cell by the largest value in the row, and then takes the standard deviation of all values in the row.

$$Metric_7 = std\left[\left.\frac{cell_1,\ cell_2,\ \cdots,\ cell_{last}}{Max\ value}\right|_{@\ 0\ Q}\right] \tag{3.7}$$

The last two regions of interest are the "low" and "high" pitch rate regions. These regions are arbitrary and will be varied (see Section 3.4.2) and all considered. As referenced before, even a pitch rate as low as 5 $\frac{deg}{s}$ includes dynamic lift effects [21]. That value and others will constitute the set of cut-off pitch rates from the arbitrary "low" to "high" magnitudes as seen in Eq. (3.8). These ranges of pitch rate will exclude the row of zero pitch rate as it is examined thoroughly in Metrics $5-7$. The range of high pitch rates will simply be the inverse of these values. If it is found that other values other than those presented in Eq. (3.8) would be best, they can be altered after the fact to produce the best metrics.

$$cut-off\ magnitudes = [5,\ 10,\ 20,\ 50]_{deg/s} \tag{3.8}$$

The same set of three metrics that were utilized for the zero pitch rate region will be used for these two regions. Put in equation form, the three metrics for a low pitch rate region are:

$$Metric_8 = \frac{\#\ of\ cells\ at\ low\ (not\ 0)\ Q\ with\ data\ points}{Total\ \#\ of\ cells\ at\ low\ (not\ 0)\ Q} \times\ 100 \tag{3.9}$$

$$Metric_9 = \frac{Total\ \#\ of\ data\ points\ at\ low\ (not\ 0)\ Q}{Total\ \#\ of\ cells\ at\ low\ (not\ 0)\ Q} \tag{3.10}$$

$$Metric_{10} = mean\left(std\left[\frac{cell_1,\ cell_2,\ \cdots,\ cell_{last}}{Max\ value}\bigg|_{@\ low\ (not\ 0)\ Q} \right]\right)_{Each\ row} \tag{3.11}$$

And similarly, the following equations depict the metrics for a high pitch rate region:

$$Metric_{11} = \frac{\#\ of\ cells\ at\ high\ Q\ with\ data\ points}{Total\ \#\ of\ cells\ at\ high\ Q} \times\ 100 \tag{3.12}$$

$$Metric_{12} = \frac{Total\ \#\ of\ data\ points\ at\ high\ Q}{Total\ \#\ of\ cells\ at\ high\ Q} \tag{3.13}$$

$$Metric_{13} = mean\left(std\left[\frac{cell_1,\ cell_2,\ \cdots,\ cell_{last}}{Max\ value}\bigg|_{@\ high\ Q} \right]\right)_{Each\ row} \tag{3.14}$$

To reiterate, metrics $8-13$ will be recursively applied to the various deviations from low to high pitch rate given in Eq. (3.8) to allow for wider explanation of why the best maneuver is the best.

Also, the overall magnitude of the above metric values will greatly depend on the step sizes used for the discretization, but this is not important as only the magnitude of the values of one maneuver with respect to the values of other maneuvers will matter. The step size of the discretization will, however, have an effects on how accurately the trace of the maneuver is mapped to regressor space, which in turn can affect the metric values. This can then alter the relative values of the outputs of the metrics from each maneuver. For example, a discretization of 10 $\frac{deg}{s}$ in pitch rate will include relatively large magnitudes as though they are in the zero pitch rate region. Conversely, a discretization that is too fine may be excluding values from the zero pitch rate region which can be practically considered zero.

This then sets the standard for defining the step size. In pitch rate, a value less than 0.1 $\frac{deg}{s}$ is essentially zero especially when considering the length of time of a maneuver is performed. In 10 s, this would only equate to a 1 deg change in AoA. The constraints in AoA are not so rigid as only metric 3 depends on this discretization, and not very strongly. Therefore, the step size for AoA is arbitrarily set at 0.1 deg increments, just for uniformity in magnitude of the discretization of both AoA and pitch rate. To summarize:

$$Q_{step\ size} = 0.1\ \frac{deg}{s} \tag{3.15}$$

$$AoA_{step\ size} = 0.1\ deg \tag{3.16}$$

The `Matlab`® code used for each of these metrics can be found in Appendix A.

The last note for the metrics defined above is to define what is "good" for each metric. For all of the metrics except for 3 and 4 (the standard deviation of the data in AoA and pitch rate), the higher the value the better. For the third and fourth maneuvers, an evenly spread data set will display a standard deviation that approaches zero. The other standard deviation values are calculated based on sections of the AoA data (for a given region of pitch rate). It is not reasonable to assume there will be data in every "column" and so an evenly spread data set will not have a Gaussian distribution, but will be very flat – a high standard deviation.

3.4 Maneuver Development

In order to verify the metrics, there needs to be a large variety of maneuvers from which to develop models. There must also be at least one (preferably more) set(s) of validation data (none of which can be used in creating any models). This way, each maneuver can be compared to the validation data and then quantified as to how well the model predicts the validation data. This process helps to accomplish both goals of this research at the same time. The metrics cannot be validated without knowing which maneuvers are best/worst and it is not known which maneuvers are best without testing them all and comparing them to validation data as described above and in Section 1.4.

The set of maneuvers to be created should include some simple baseline maneuver with a very basic motion which can be similar to an actual flight test maneuver. The current best maneuver (the chirp sinusoidal motion) should also be included so the chopped sinusoidal motion can be compared directly to it. Having these two maneuvers, knowing the latter is better than the former, will help to give some initial results from the metrics by seeing which parameters are better in the chirp motion than the basic.

Other than the two maneuvers discussed above, any number of variations of the chopped sinusoidal motion can be included. It is also wise to include some maneuvers whose purpose is to optimize certain aspects of the metrics for their validation and not necessarily focused on producing the best model (which is the primary goal of the other maneuvers). Since the maneuvers will focus on the chopped sinusoid, they may all look very similar with primary differences in how they blend static and dynamic regions.

Before the maneuvers are presented, the methodology for creating a maneuver should be discussed. Equation (3.17) is a generic function for describing a sinusoidal type motion where initial and final amplitudes and frequencies can be set as well as how quickly the transition takes place from initial to final states. The output of this equation can be used as the set of roll, pitch, or yaw angles over time, which will be used for the AoA variation for the purposes of these maneuvers.

$$S(t) = \tilde{s}(\hat{t}) \cos[2\pi(\beta_f \hat{t}^{1+\lambda_f} + f_1 \hat{t} + \Phi/360)] \tag{3.17}$$

with

$$\tilde{s}(\hat{t}) = \beta_a \hat{t}^{1+\lambda_a} \tag{3.18}$$

$$\beta_a = \frac{a_2 - a_1}{(t_{max} - t_0)^{\lambda_a}} \tag{3.19}$$

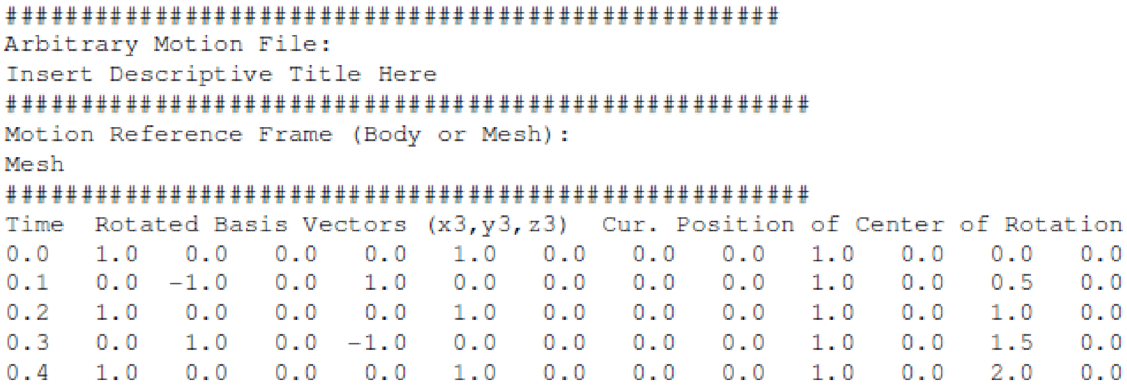
$$\hat{t} = (t - t_0) \tag{3.20}$$

$$\beta_f = \frac{f_2 - f_1}{(t_{max} - t_0)^{\lambda_f}} \tag{3.21}$$

where t is the actual time, t_{max} is the maximum time of the maneuver, t_0 is the start time of the maneuver, a_2 and f_2 are the final amplitude and frequency, a_1 and f_1 are the initial amplitude and frequency, λ_f and λ_a are the frequency and amplitude shift parameters that define the variation from initial to final amplitude and frequency states, and finally Φ is the phase shift in degrees. Note that an λ_f or λ_a equal to 1.0 gives a linear variation from f_1 to f_2 and a_1 to a_2, respectively [12].

Once the maneuvers are created, they are introduced into the CFD solver by the means of an arbitrary input file. The Kestrel User Guide [12] describes the format this file should have. There are eight header lines, followed by 13 columns of data. The first column is current time (seconds), the next three columns are the basis vector, n_x, of the newly rotated x-axis in terms of the initial, inertial coordinate system, likewise, the next three columns are the basis vector, n_y, of the rotated y-axis, the next three are the basis vector, n_z, of the rotated z-axis, and the last three columns are the (x, y, z) coordinates to the instantaneous center of rotation. Figure 3.2 show a simple example of this type of file and the `Matlab`® code for the creation of this file can be found in Appendix D.

3.4.1 The Final Maneuvers. The first maneuver will be a simple sinusoidal motion with constant frequency and linearly varying amplitude. This type of motion is inspired by the type of motion that occurs when a disturbance enters the system and an under-damped response ensues. The magnitude and subsequent oscillations of this maneuver are much higher and longer than would be seen in a normal aircraft,



```
####################################################
Arbitrary Motion File:
Insert Descriptive Title Here
######################################################
Motion Reference Frame (Body or Mesh):
Mesh
######################################################
Time  Rotated Basis Vectors (x3,y3,z3)  Cur. Position of Center of Rotation
0.0   1.0   0.0   0.0   0.0   1.0   0.0   0.0   0.0   1.0   0.0   0.0   0.0
0.1   0.0  -1.0   0.0   1.0   0.0   0.0   0.0   0.0   1.0   0.0   0.5   0.0
0.2   1.0   0.0   0.0   0.0   1.0   0.0   0.0   0.0   1.0   0.0   1.0   0.0
0.3   0.0   1.0   0.0  -1.0   0.0   0.0   0.0   0.0   1.0   0.0   1.5   0.0
0.4   1.0   0.0   0.0   0.0   1.0   0.0   0.0   0.0   1.0   0.0   2.0   0.0
```

Figure 3.2: Sample arbitrary motion input file [12]

but this selection is a good introductory motion. For this type of maneuver, the initial pitch rate will be the highest at the very beginning as that is where the highest amplitude change happens (this being true only for the constant frequency case). The amplitude, rate of decrease of the amplitude, and initial AoA, will all be set to provide a maximum AoA of 30 *deg* and a minimum amplitude of -10 *deg* (as all of the maneuvers will attempt to do), and a final amplitude of 0 *deg*. The final inputs to Eq. (3.17) are listed in Table 3.1. These inputs produce the motion seen in Figure 3.3 which covers the regressor space seen in Figure 3.4. Figure 3.5 shows the pitch rate of this maneuver over time along with AoA. This maneuver has a very even distribution of the regressor space but not as much overall coverage as other maneuvers do.

Table 3.1: List of Inputs to Eq. (3.17) for each maneuver.

Maneuver # (s)	AoA_i	a_1	a_2	f_1	f_2	λ_a	λ_f	Φ
1	8.0	22.5	0.0	0.8	0.8	1.0	1.0	-90
2	8.0	22.5	0.0	0.75	1.6	1.0	1.9	-90
3 - 7	8.0	27.0	0.0	0.6	1.35	1.0	1.9	-90

One of the maneuvers that must be included is the current best, the chirp sinusoid, in order to determine if any of the chopped sinusoid maneuvers can better predict the validation data. The chirp sinusoidal maneuver is very similar to the first maneuver except the frequency is also allowed to be varied. This produces a maneuver which utilizes the maximum pitch rate much more in order to cover a wider range of

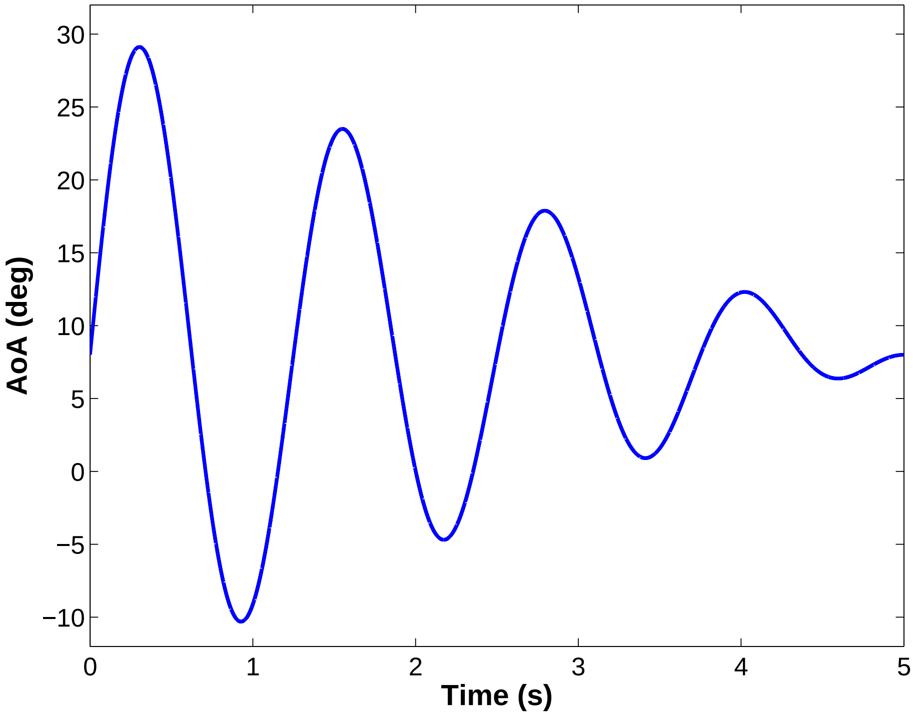


Figure 3.3: Motion of first maneuver.

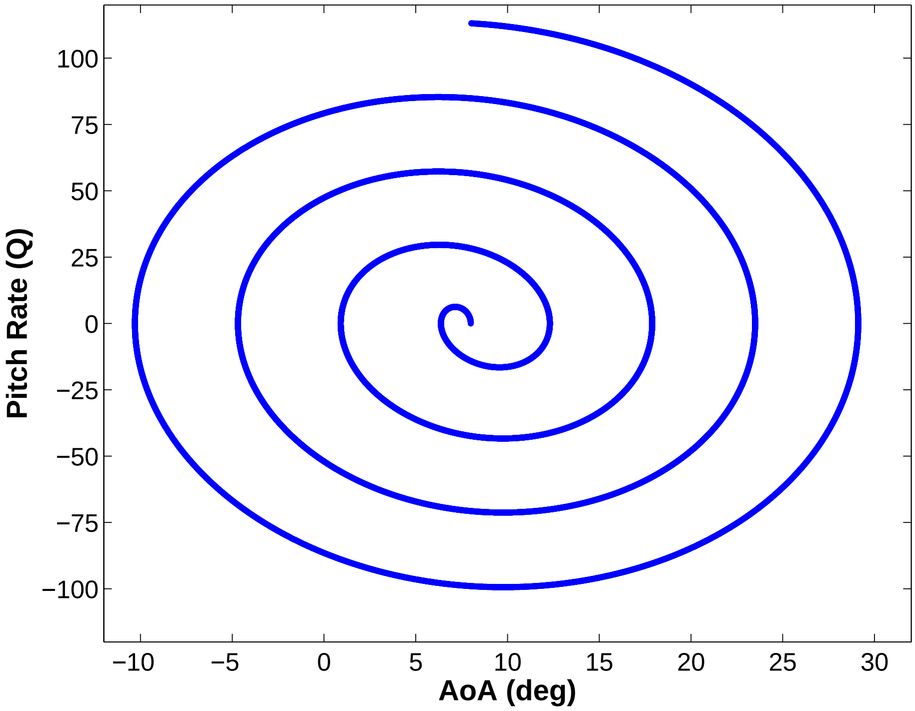


Figure 3.4: Regressor space covered by the first maneuver.

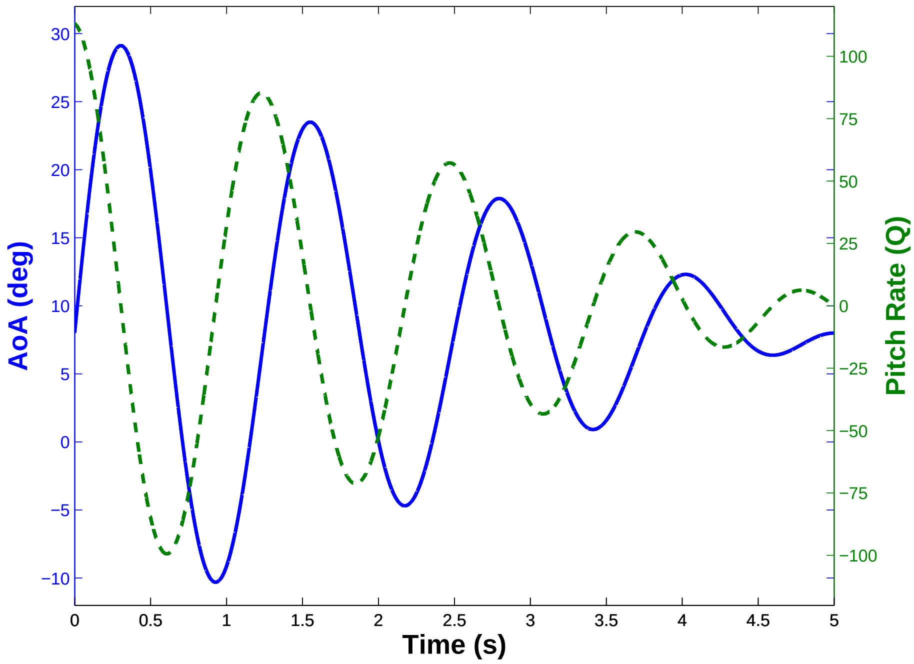


Figure 3.5: Pitch rate of first maneuver compared to AoA.

AoA in the same amount of time. The use of the maximum pitch rate at the peak of a given cycle is not held for the entire maneuver as that would produce too much "repeat" data at the low AoA values as this is where the maximum pitch rate occurs for a given cycle in a sinusoidal motion. By decreasing the maximum pitch rate, the center region of the regressor space is filled in more evenly as seen in Figure 3.7.

The inputs for Eq. (3.17) are listed in Table 3.1 along with the rest of the maneuvers. The same three set of figures shown for the first maneuver are presented for this maneuver in Figure 3.6, Figure 3.7, and Figure 3.8. A linear decrease from the initial to the final AoA was kept as it provides a very even coverage of the regressor space as seen in Figure 3.7.

The rest of the maneuvers will each require more discussion as the implementation of a static (or another type) region requires manipulation beyond what is provided from Eq. (3.17). The third maneuver will introduce the largest static regions of all the maneuvers in attempt to get reliable static data and to also maximize the metrics for zero pitch rate (metrics $5-7$). For this purpose, roughly a quarter of a second

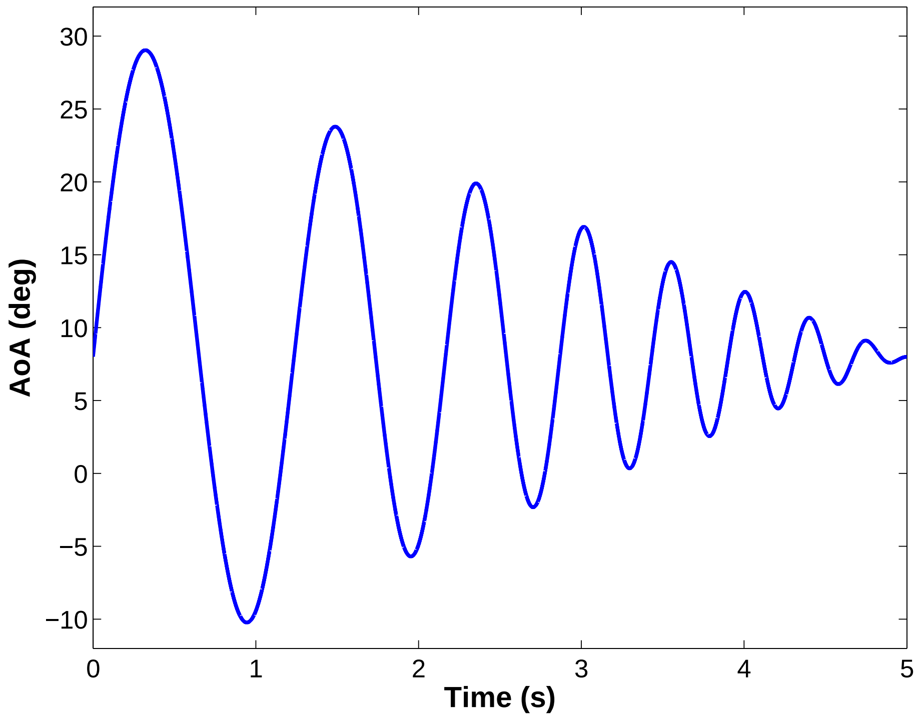


Figure 3.6: Motion of second maneuver.

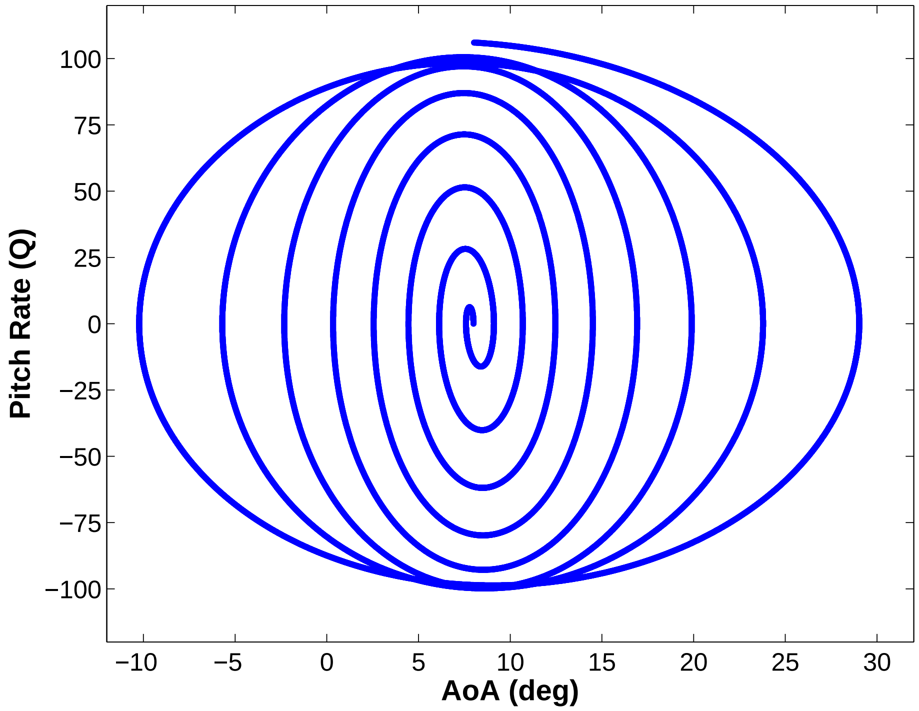


Figure 3.7: Regressor space covered by the second maneuver.

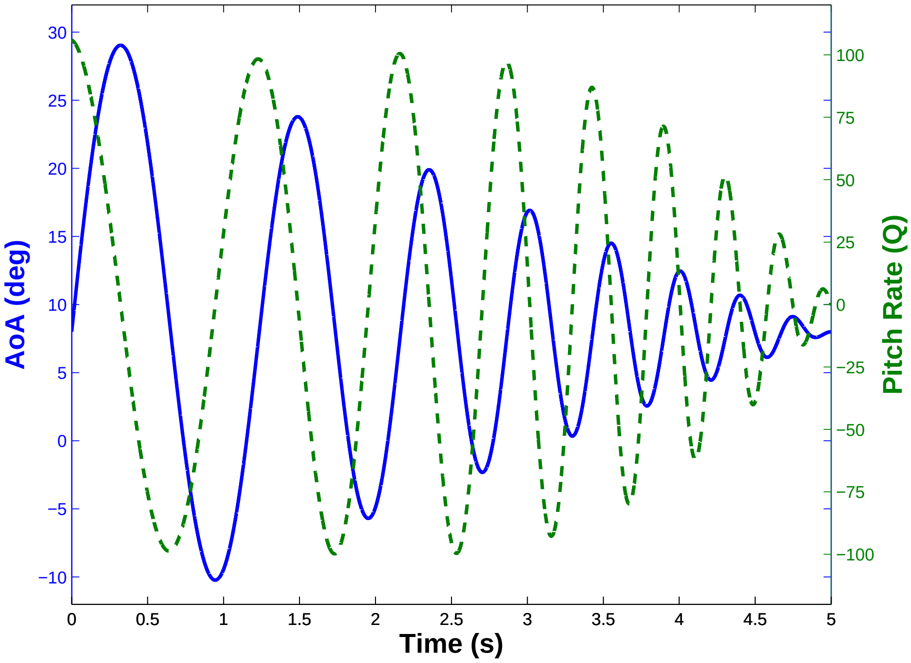


Figure 3.8: Pitch rate of second maneuver compared to AoA.

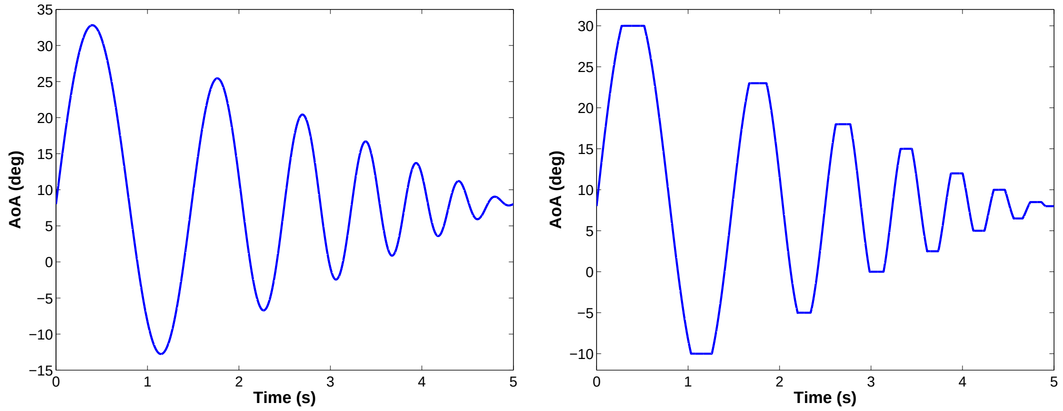
(about 4 to 5 percent of the total maneuver) of data will be held at zero pitch rate for each peak amplitude through the maneuver as seen in Figure 3.11. The long length of time is to account for the unsteadiness of the flow at large AoA values, and the static regions are shorter at the lower AoA values.

Maneuver 3 poses the trade-off required when implementing static regions into a maneuver. The second maneuver, although without static data, is able to produce more oscillations (Figure 3.6) which in turn provides better overall coverage (Figure 3.7) than the third maneuver (see Figure 3.11 and Figure 3.12, respectively). The results will have to be examined before a conclusion of how big of an effect the lesser coverage of the regressor space will have on accurately modeling the dynamic response of the aircraft for the chopped sinusoidal maneuvers.

This third maneuver is created by starting with a sinusoidal motion which overshoots the bounds of maximum and minimum AoA as seen in Figure 3.9(a). The input values for Eq. (3.17) are seen in Table 3.1 which produce Figure 3.9(a). The array of AoA for that motion is then run through a loop which “chops” off the peak amplitudes at specified values (Eq. (3.22)) which produces Figure 3.9(b).

$$Chop\ Values = \begin{bmatrix} 30 & 23 & 18 & 15 & 12 & 10 & 8.5 \\ \text{-10} & \text{-5} & 0 & 2.5 & 5 & 6.5 & 8 \end{bmatrix} \tag{3.22}$$

When this is done, however, sharp corners are left on the plot as seen in Figure 3.10(a). These corners can be a source of numerical instability for the CFD solver so they must be rounded. To do so, the slope is found a few points before the chop occurs. This then provides a known $(x_1,\ y_1)$ at this point and its corresponding slope, s_1. The final y_2 value is also known along with its slope of zero ($s_2 = 0$). That is enough information to fully define the equation for a circle at that point. This process is given by the following set of equations:



(a) Before

(b) After

Figure 3.9: Initial motion of third maneuver before and after "chopping."

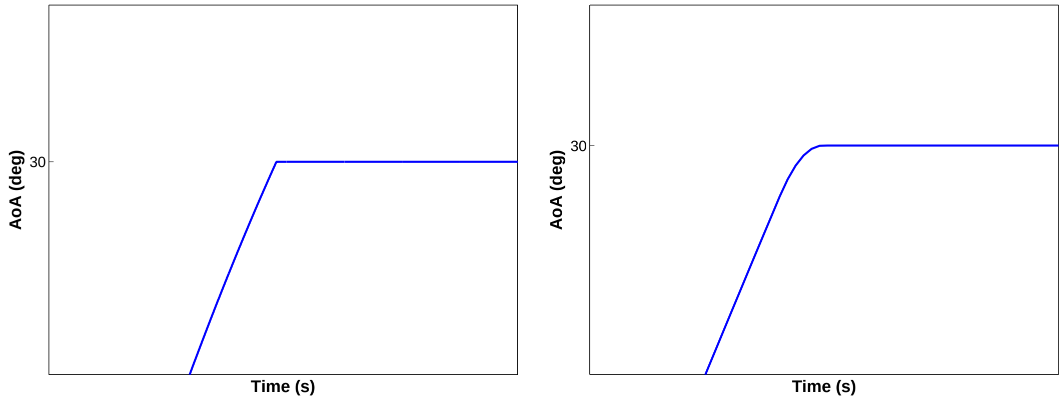
$$x_0 = x_1 \pm \frac{(y_2 - y_1)s_1}{\sqrt{s_1^2 + 1} - 1} \tag{3.23}$$

$$y_0 = y_1 \pm \frac{(x_1 - x_0)}{s_1} \tag{3.24}$$

$$r = y_2 - y_0 \tag{3.25}$$

$$y = \sqrt{r^2 - (x - x_0)^2} + y_0 \tag{3.26}$$

Where $(x_0,\ y_0)$ is the location of the origin of the circle, r is the radius, and y is the value of the circle at a given x location. Also, the ± sign determines if you are on the "upper" or "lower" surface of the circle. The `Matlab`® code for this process can be found in Appendix C. After implementing this process, the sharp corner is transformed into what is seen in Figure 3.10(b).



(a) Before

(b) After

Figure 3.10: Corners of third maneuver before and after smoothing sharp corners from being chopped

Figure 3.11 shows the final form of the third maneuver with Figure 3.12 and Figure 3.13 showing its coverage of the regressor space and pitch rate versus AoA over time respectively. Figure 3.12 shows sparse amounts of data in the approximately $\pm 40\ \frac{deg}{s}$ pitch rate range, but the figure does not show the large number of data points at exactly zero pitch rate for each AoA value at the zero crossings.

The rest of the maneuvers start with the same two motions shown in Figure 3.9. The difference between the next two maneuvers from the third is the treatment of the blending from dynamic to static data. Both will include low pitch rate oscillations at the corners of each chopped region. This low pitch rate will spread some of the zero pitch rate data into some of the "bare" areas seen in Maneuver 3's regressor space (Figure 3.12). Maneuver 4 will stay conservative and keep a fairly large static region

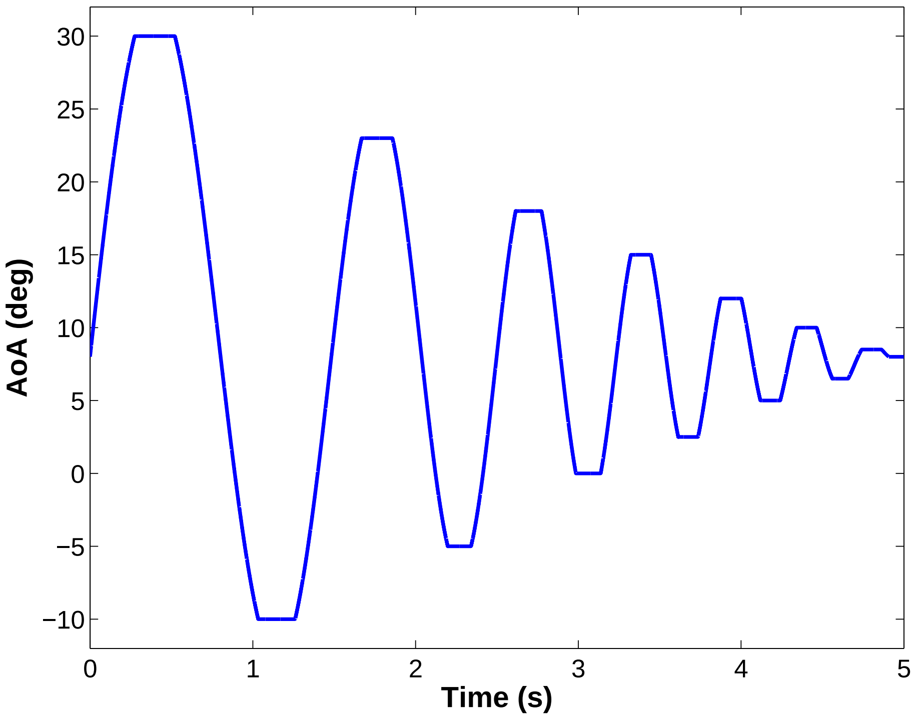


Figure 3.11: Motion of third maneuver.

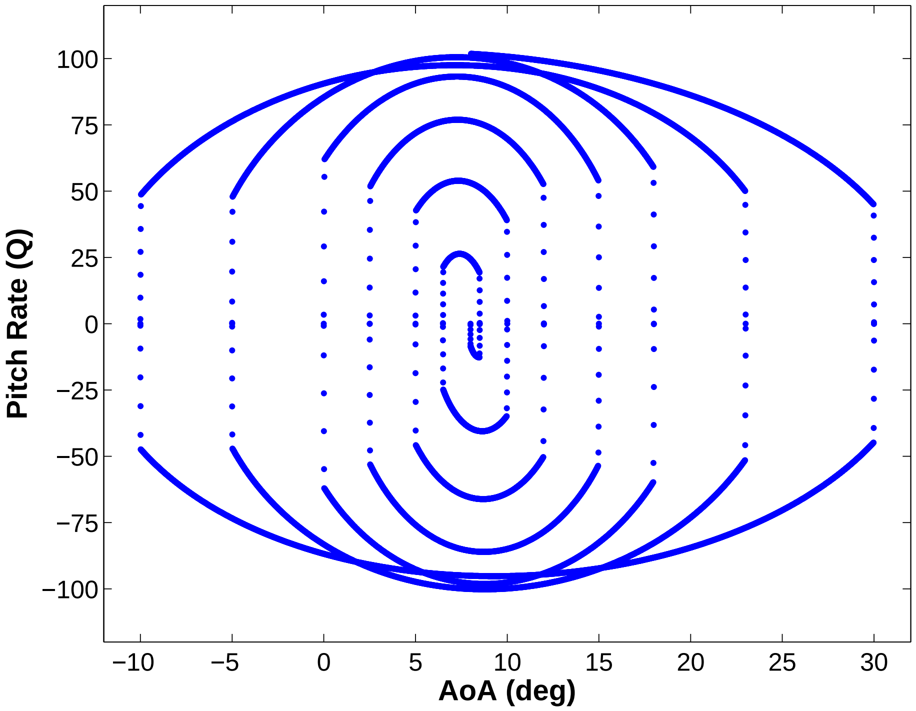


Figure 3.12: Regressor space covered by the third maneuver.

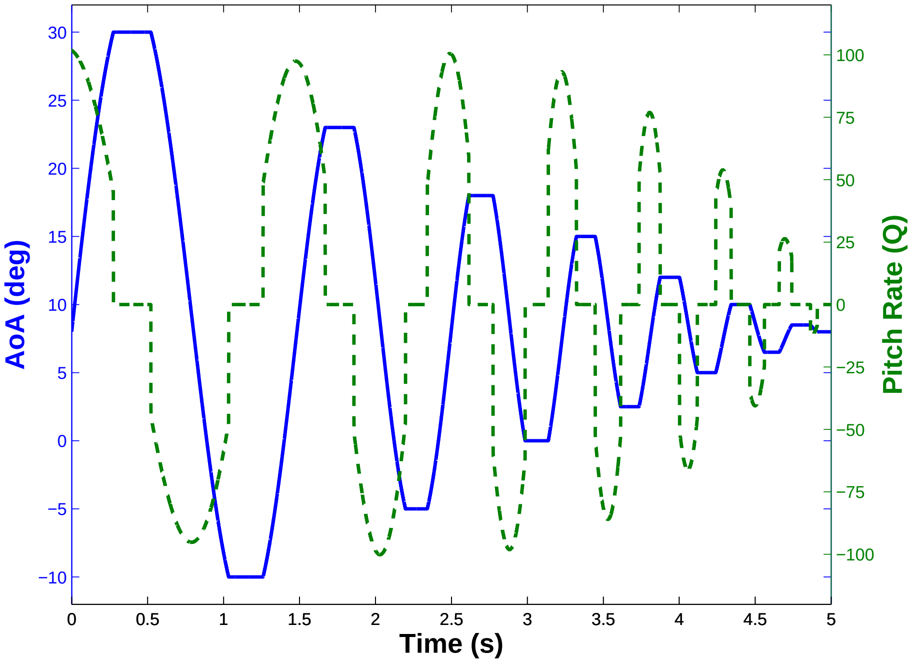


Figure 3.13: Pitch rate of third maneuver compared to AoA.

while Maneuver 5 will have more low pitch rate oscillations with fewer truly static points.

To do this, a cubic spline interpolation scheme will be used to smooth out the motion seen in Figure 3.9(b). The built-in `Matlab`® “spline” function will be used. This spline function solves for a piecewise polynomial for every region in between each data point. This way, the function will exactly predict every point used for the interpolation. At sharp corners, however, an overshoot will occur followed by a damped oscillation – producing the low-pitch rate oscillations that are desired [28].

Since the spline function exactly predicts every point input into the function, the data from Figure 3.9(b) must first have data points removed at specified intervals. These values are then input into the function and sampled at the original input values. This process is best explained through an example; assume a data set has the following original values:

$$\begin{Bmatrix} x \\ y \end{Bmatrix} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 \\ 24 & 25 & 26 & 27 & 28 & 29 & 30 & 30 & 30 & 30 & 30 & 30 & 30 \end{bmatrix} \tag{3.27}$$

This example data set can then be be sampled at every fourth value to produce:

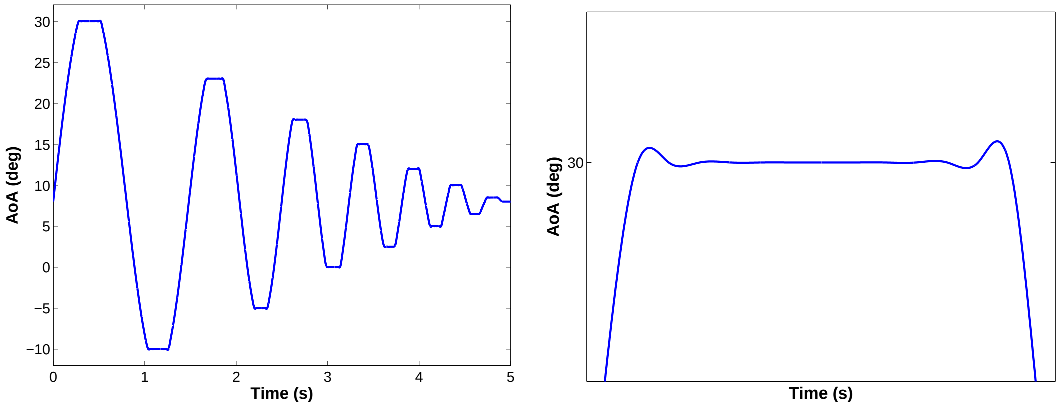
$$\begin{Bmatrix} x_s \\ y_s \end{Bmatrix} = \begin{bmatrix} 1 & 5 & 9 & 13 \\ 24 & 28 & 30 & 30 \end{bmatrix} \tag{3.28}$$

The sampled data, (x_s, y_s), is then fed into the spline function as the "known values" and the original x values are input as the desired output interpolated values calculated by the spline function. This example is performed in `Matlab`® and the following result are produced:

$$\begin{Bmatrix} x_f \\ y_f \end{Bmatrix} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 \\ 24 & 25.2 & 26.3 & 27.2 & 28 & 28.7 & 29.3 & 29.7 & 30 & 30.2 & 30.3 & 30.2 & 30 \end{bmatrix} \tag{3.29}$$

It is seen that the maximum value of the final y_f values is above the maximum value of the initial data set, y. With a much longer data set and a wider sampling rate, this overshoot will be more pronounced and will be followed by undershooting the final static data and then overshooting, etc., until the response is damped out entirely.

Maneuver 4 takes the original data set and samples it at every $0.02/dt$ intervals. The final time step, dt, will be discussed in Section 3.5, but will be on the order of $1E-4$ which will produce a sampling rate on the order of $1E2$. Performing these steps results in the final maneuver as seen in Figure 3.14(a) with a zoomed-in look at the first spline blending from static to dynamic regions Figure 3.14(b). This maneuver covers the regressor space seen in Figure 3.15 and has the pitch rates over time seen in Figure 3.16. The regressor space looks "wiggly" which results in more overall coverage and it can be vaguely seen that the there are over-lapping regions around zero pitch rate with small deviations of AoA. This should produce quality low and zero pitch



(a) Total Maneuver.

(b) Zoomed in view of first static region.

Figure 3.14: Fourth maneuver.

rate data at these specific AoA values. This regressor space should also be compared to the regressor of the third maneuver (Figure 3.12). The large spaces seen in the third regressor space are not apparent in the fourth.

The fifth maneuver is very similar to the fourth except the low pitch rate is extended further into the static regions by increasing the sampling time to $0.045/dt$ intervals, increasing the sampling time by 2.25 times. Plots similar to the fourth maneuver can be seen in Figure 3.17, Figure 3.18, and Figure 3.19 respectively. The regressor space shows smooth transitions from dynamic to static data and there is a larger range of the over-lapped low pitch rate data around the static AoA values. This larger spread in low pitch rate data at the angles means there are fewer data points at exactly zero pitch rate, however. This will produce the main difference in Maneuvers 4 and 5 – the fourth maneuver is more conservative in the sense that it has more zero-pitch rate data points (including the low pitch rate points).

Two final maneuvers are to be examined. The primary purpose of these is to produce an optimization of the importance of the low pitch rate metrics as well as to provide greater AoA coverage at low pitch rate data as compared to Maneuvers 4 and 5 whose low pitch rate data stayed near the static values. Similarly, Maneuver 3

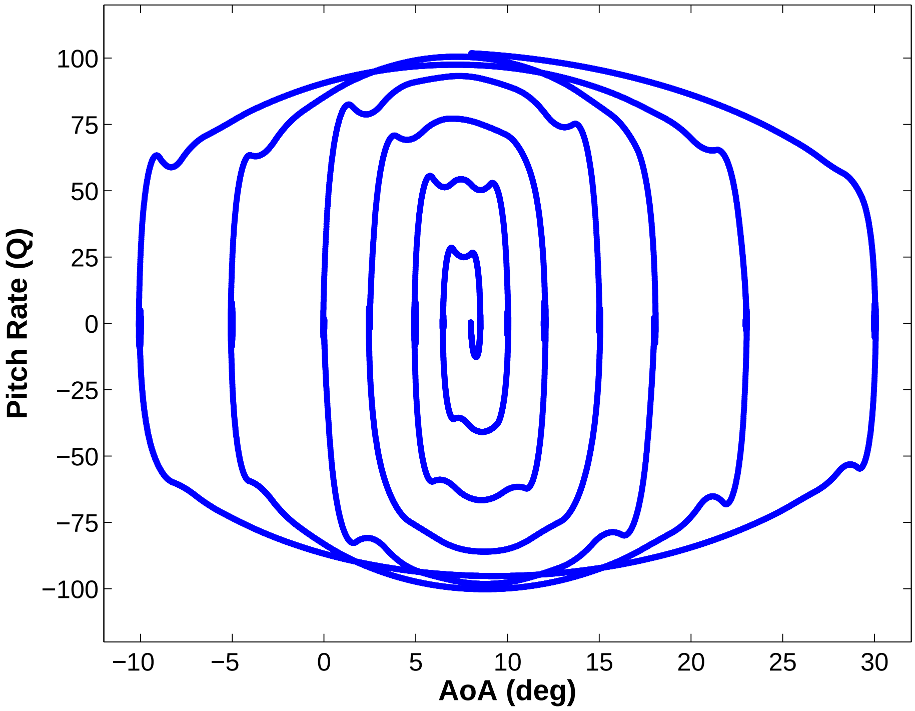


Figure 3.15: Regressor space covered by the fourth maneuver.

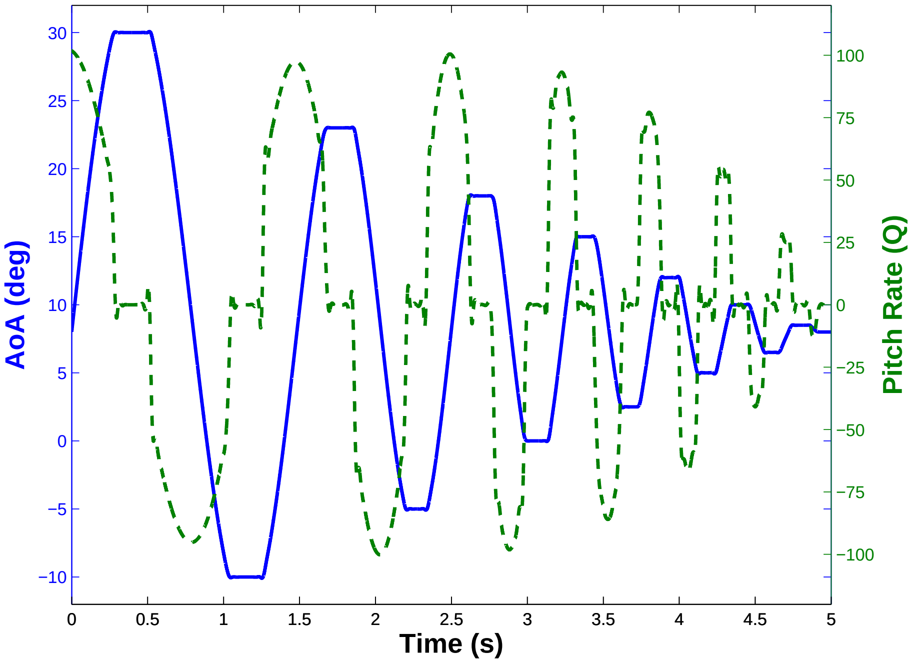
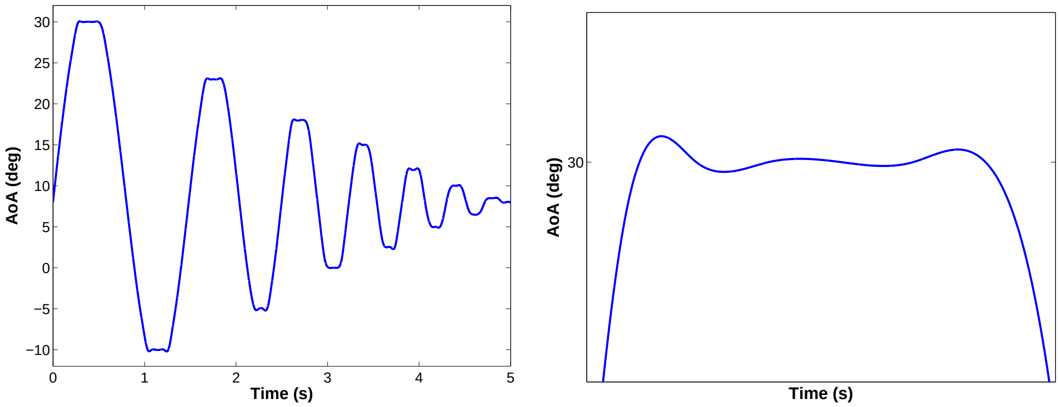


Figure 3.16: Pitch rate of fourth maneuver compared to AoA.



(a) Total Maneuver.

(b) Zoomed in view of first static region.

Figure 3.17: Fifth maneuver.

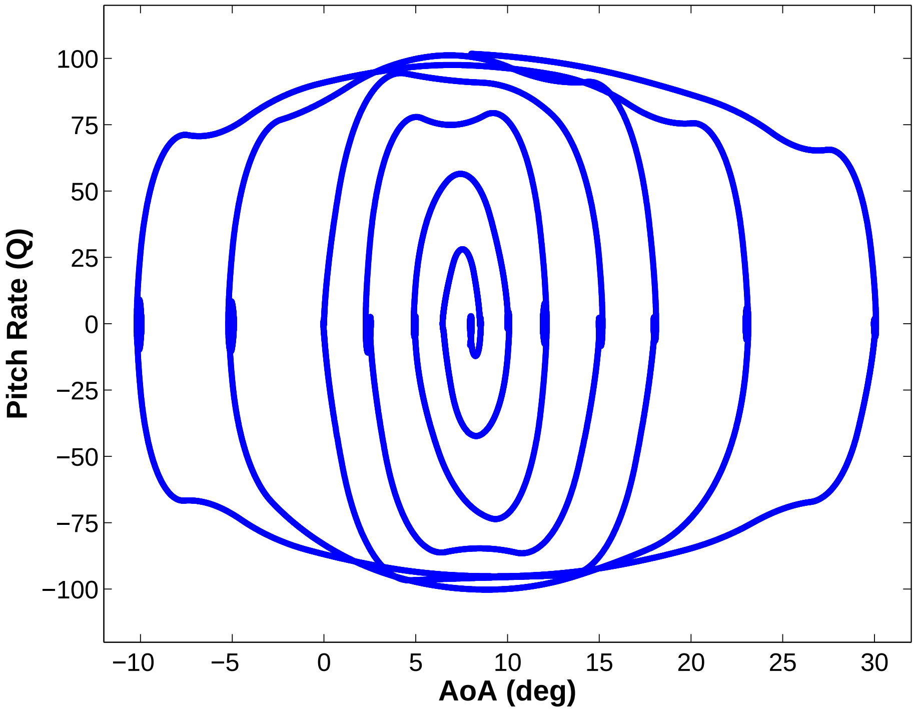


Figure 3.18: Regressor space covered by the fifth maneuver.

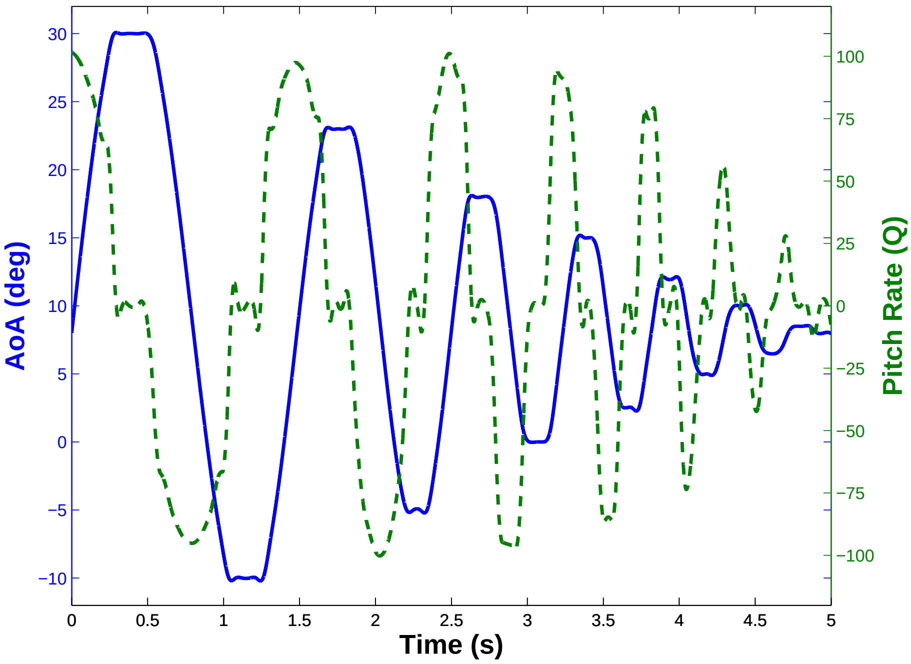


Figure 3.19: Pitch rate of fifth maneuver compared to AoA.

is essentially the optimization of the zero pitch rate data metrics – to help define the importance of those metrics with respect to others.

For these maneuvers, all static data will be removed in place of secondary low-amplitude sinusoidal motion. This will also help to magnify (or minimize) the importance of purely static data over just low pitch rate data. To do this, the maneuver found in Figure 3.9(a) is again utilized which is then chopped to include the static regions, but this time the cutoff values are altered to be:

$$Chop\ Values_6 = \begin{bmatrix} 29.75 & 23 & 18 & 15 & 12 & 10 & 8.5 \\ \text{-9.75} & \text{-5} & 0 & 2.5 & 5 & 6.5 & 8 \end{bmatrix} \tag{3.30}$$

Where the first column houses the only changes which are made to keep the final maneuver within the regressor space. The output looks nearly identical to Figure 3.9(b). From here, each static region is now replaced with a low amplitude sinusoidal motion with a specified maximum pitch rate of 20 $\frac{deg}{s}$. This will produce many data points all within this region and very few static data points for the reasons specified above.

This calculation is done with the use of a simple sine function. The base function being:

$$AoA = A\sin(\omega t) \tag{3.31}$$

With A being the amplitude, ω the frequency, and t the instantaneous time. The derivative can then be taken to solve for the pitch rate over time:

$$Q = \frac{\partial AoA}{\partial t} = A\omega\cos(\omega t) \tag{3.32}$$

Equation (3.32) gives the means to specify a pitch rate over time. The derivative of this equation, set equal to zero, will solve for the set of ωt values that produce the maximum pitch rates. This is seen in the following:

$$Q_{max} = \frac{\partial Q}{\partial t} = 0 = -A\omega^2\sin(\omega t) \Rightarrow \sin(\omega t) = 0 \Rightarrow \omega t = [0,\ \pi,\ 2\pi, \cdots] \tag{3.33}$$

Substituting the set of values of ωt given in to Eq. (3.32) produces for all variations of ωt:

$$Q|_{max\ Q} = A\omega \tag{3.34}$$

So far, both A and ω are undefined. The frequency is directly proportional to the period of the wavelength, however, and the total time length, Δt, of each static region is known. The number of cycles only needs to be set in order to define ω. The number of cycles must include a half cycle in order to allow for a smooth transition from the dynamic data before the static data and then the dynamic data after. In other words, the pitch rate changes sign from one side of the static data to the other so a half cycle must be included to account for this variation.

A half cycle would only cover AoA values higher in magnitude than the static data, but will include the largest range in AoA as is desired for these maneuvers. One and a half cycles will include AoA values higher and lower than the static data and although the amplitudes of AoA will be smaller, they will be larger than any higher number of cycles. For this reason, one and a half cycles per static region is chosen. The period, T, of the cycle can then be calculated from this information, which in turn allows for calculation of the frequency and then utilizing the defined maximum pitch rate allows for calculation of the amplitude, A. This flow of calculations is seen in the following:

$$T = \frac{\Delta t}{1.5} \tag{3.35}$$

$$\omega = \frac{2\pi}{T} \tag{3.36}$$

$$A = \frac{Q_{max}}{\omega} \tag{3.37}$$

Note that the maximum pitch rate, Q_{max} is referred to in $\frac{deg}{s}$ and the frequency is in $\frac{rad}{s}$ so the appropriate unit conversion must be done.

All of the inputs for Eq. (3.31) are now known and the desired low pitch rate oscillations can be implemented, where the sign of the function must be switched for the static regions on the "bottom" of the maneuver (see code in Appendix C). This will still result in a discontinuity at the interface of the large and small amplitude AoA variations. The spline technique discussed for Maneuvers 4 and 5 will be implemented to dissolve this problem but with a sampling rate of every $0.015/dt$ iterations to alter the final maneuver less. The sixth maneuver along with its regressor coverage and pitch rate can be seen in Figure 3.20, Figure 3.21, and Figure 3.22 respectively.

The seventh maneuver is very similar to the sixth except the maximum pitch rate is raised to 50 $\frac{deg}{s}$ to erase the white spaces seen between $\pm$ 20 to $\pm$ $50\frac{deg}{s}$ in Figure 3.21. This will also allow for larger AoA variations within this region which provides better coverage of the regressor space in this range. Also, the chop values for the static regions are changes again to keep the maneuver within the desired regressor

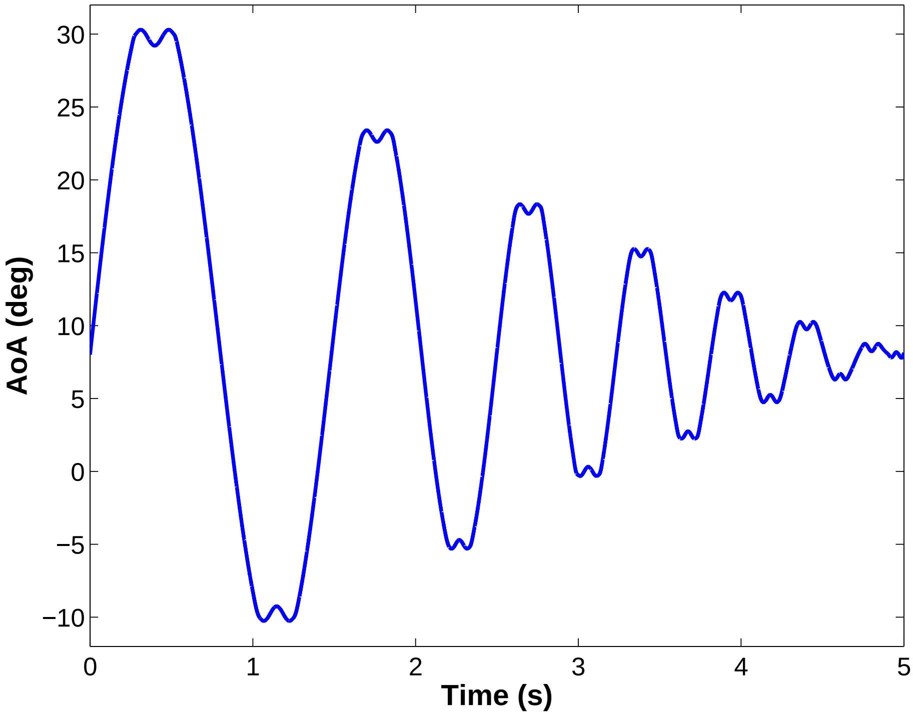


Figure 3.20: Motion of sixth maneuver.

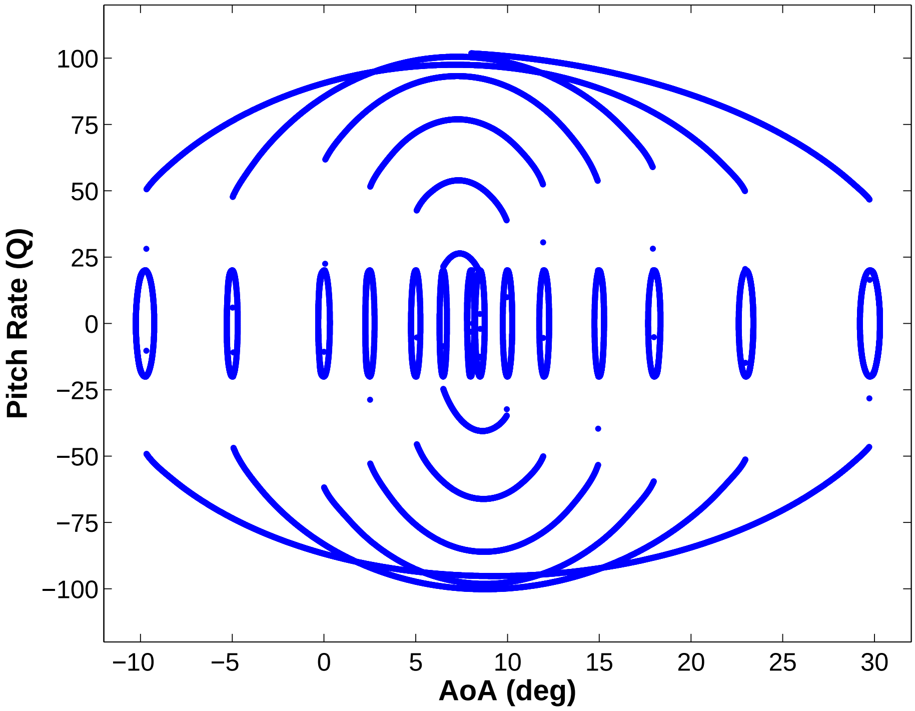


Figure 3.21: Regressor space covered by the sixth maneuver.

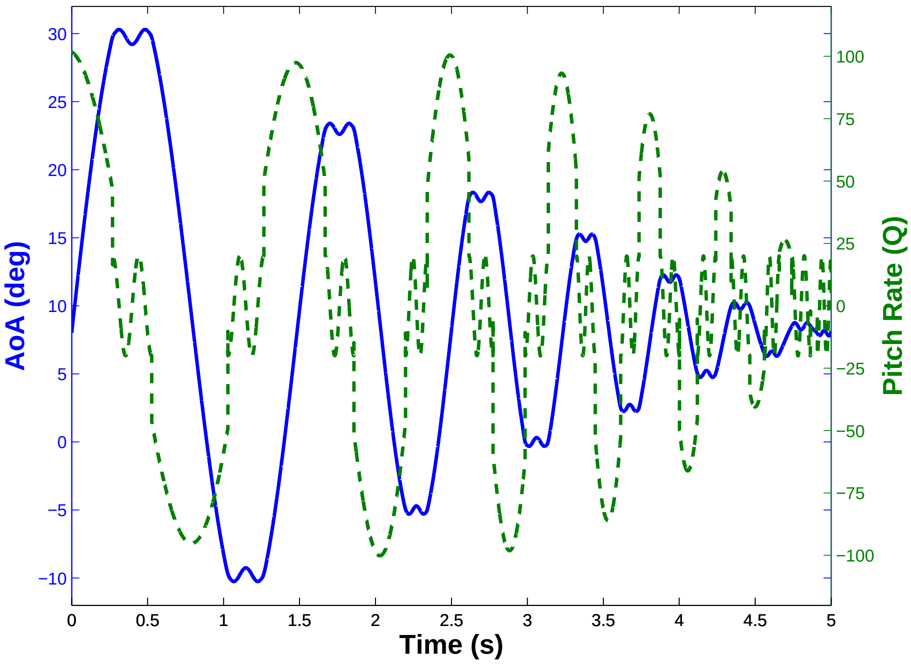


Figure 3.22: Pitch rate of sixth maneuver compared to AoA.

space as was done in Maneuver 6. The chop values are:

$$Chop\ Values_7 = \begin{bmatrix} 29 & 23 & 18 & 15 & 12 & 10 & 8.5 \\ \text{-9} & \text{-5} & 0 & 2.5 & 5 & 6.5 & 8 \end{bmatrix} \tag{3.38}$$

Producing the plots similar to the rest of the maneuvers as seen in Figure 3.23, Figure 3.24, and Figure 3.25 respectively. The regressor space coverage in the $\pm\, 50\frac{deg}{s}$ region for Maneuver 7 is the best of all maneuvers although there is virtually no static data which is the primary focus of most of the maneuvers.

3.4.2 Outputs from Metrics. The maneuvers have now all been defined as well as the metrics. These seven maneuvers can now each be fed through the metrics to see what relative values are presented for each. The first seven metrics are performed only once for each maneuver while the last six metrics will each be performed four times as specified in Section 3.3.1 and seen in Eq. (3.8). For the purpose of these metrics, a time step of 0.0002 s is used.

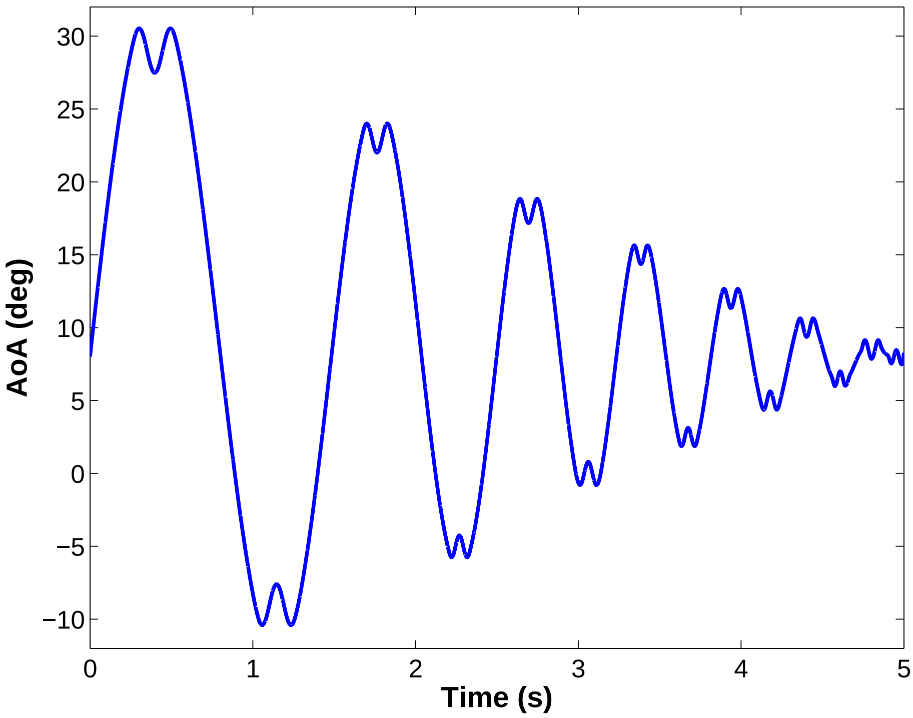


Figure 3.23: Motion of seventh maneuver.

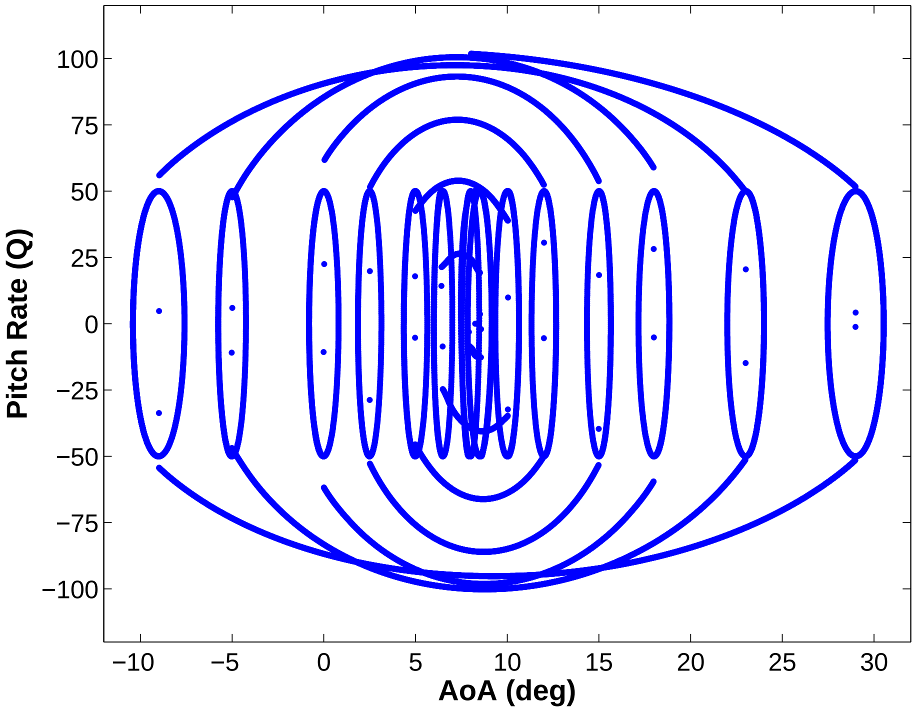


Figure 3.24: Regressor space covered by the seventh maneuver.

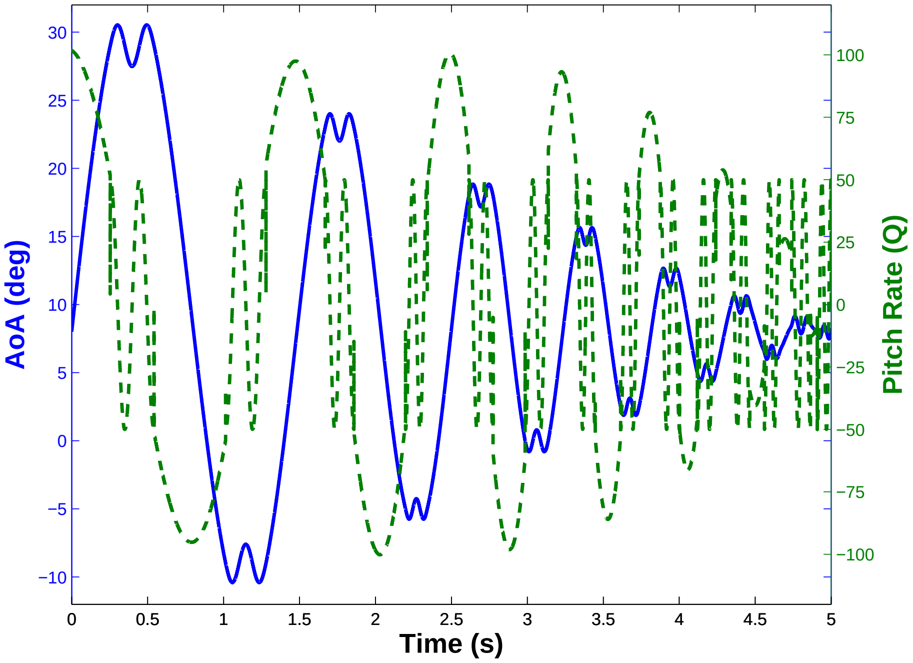


Figure 3.25: Pitch rate of seventy maneuver compared to AoA.

Before the results of the metrics are output, the coverage of the discretized regressor space as stated earlier and seen in Figure 3.1, will be displayed for each maneuver. A note must be made about the fineness of the discretization in the plots shown; the step size shown is ten times the step size used for the actual metric calculations. If the actual step sizes are used, the plots would be too small to see. It should also be noted the z-axis range is not on the same scale for Figures 3.26 - 3.32. These surface plots help to illustrate the coverage of the discretized regressor spaces as well as aid in understanding the results of the metrics.

In all of the tables shown, a green cell indicates one of the best values for a given metric, a red cell one of the worst, and the white are the median values. These colors make for easy indication of what a maneuver's strengths and weaknesses are in the metrics.

Table 3.2 shows very poor metrics for the first maneuver except in the "evenness" of the maneuver in which it does quite well. The other metrics at various pitch rate

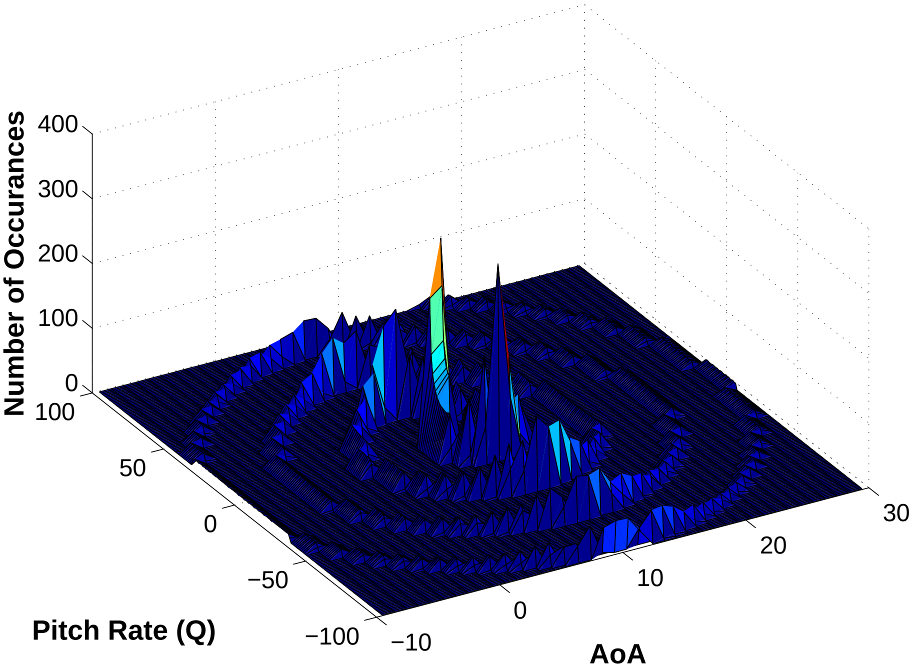


Figure 3.26: Discretized regressor space for first maneuver

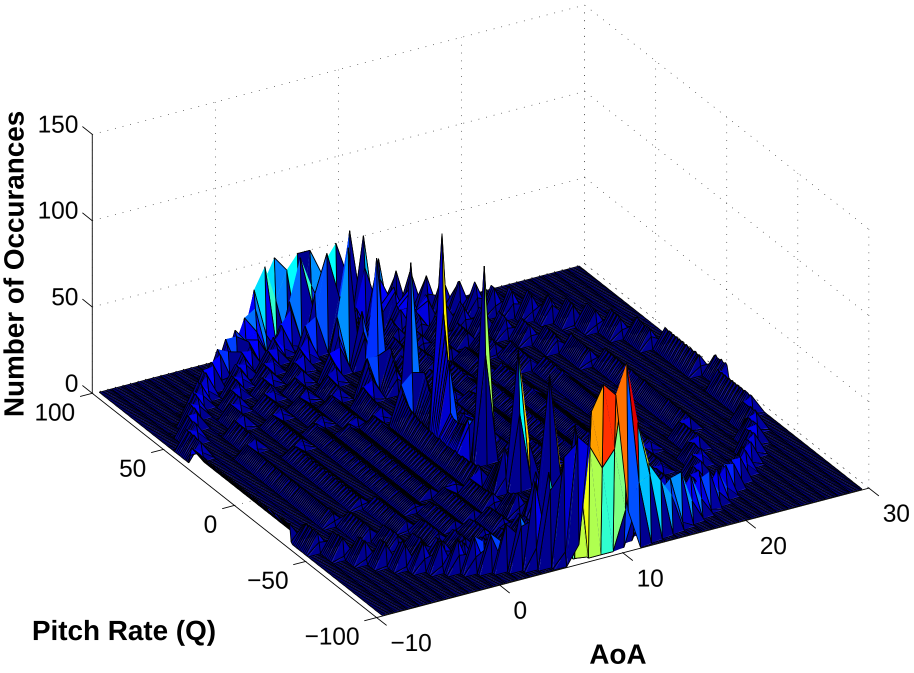


Figure 3.27: Discretized regressor space for second maneuver

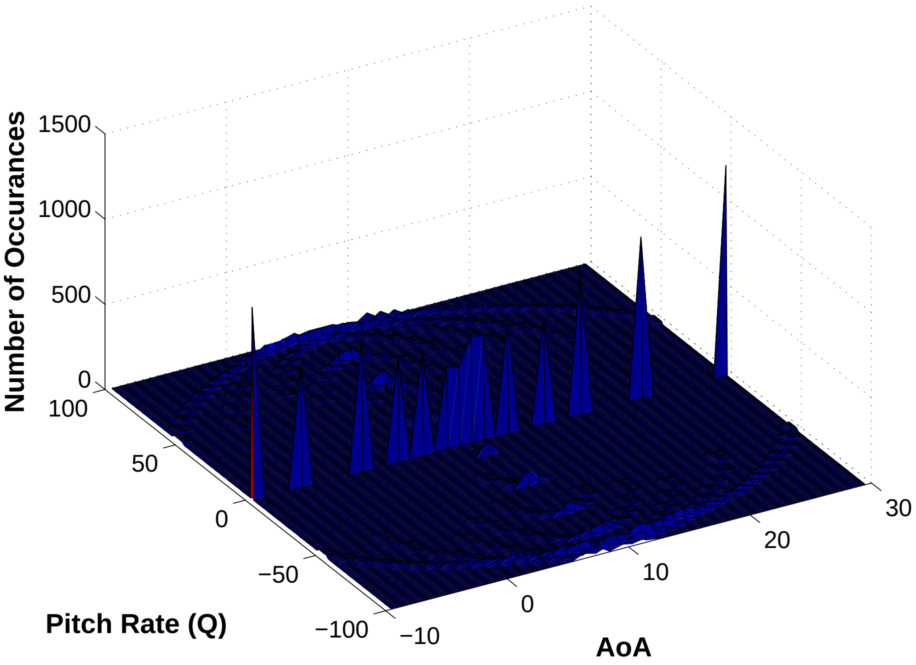


Figure 3.28: Discretized regressor space for third maneuver

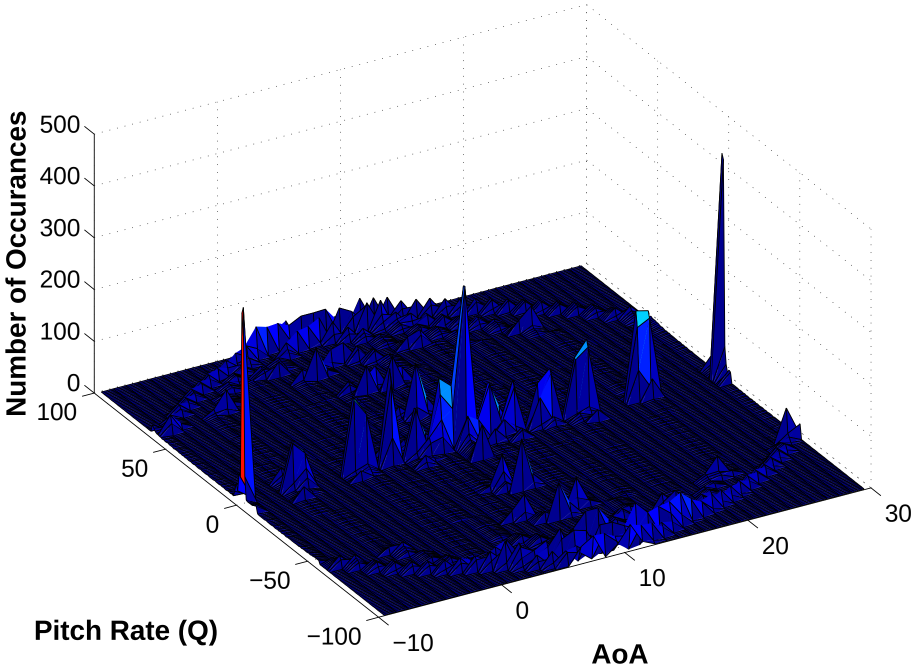


Figure 3.29: Discretized regressor space for fourth maneuver

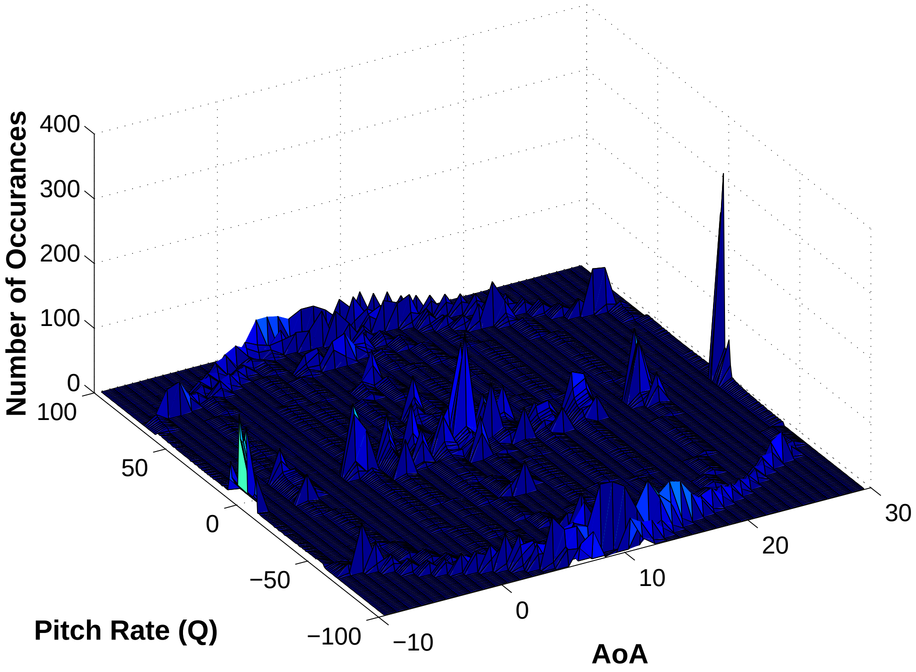


Figure 3.30: Discretized regressor space for fifth maneuver

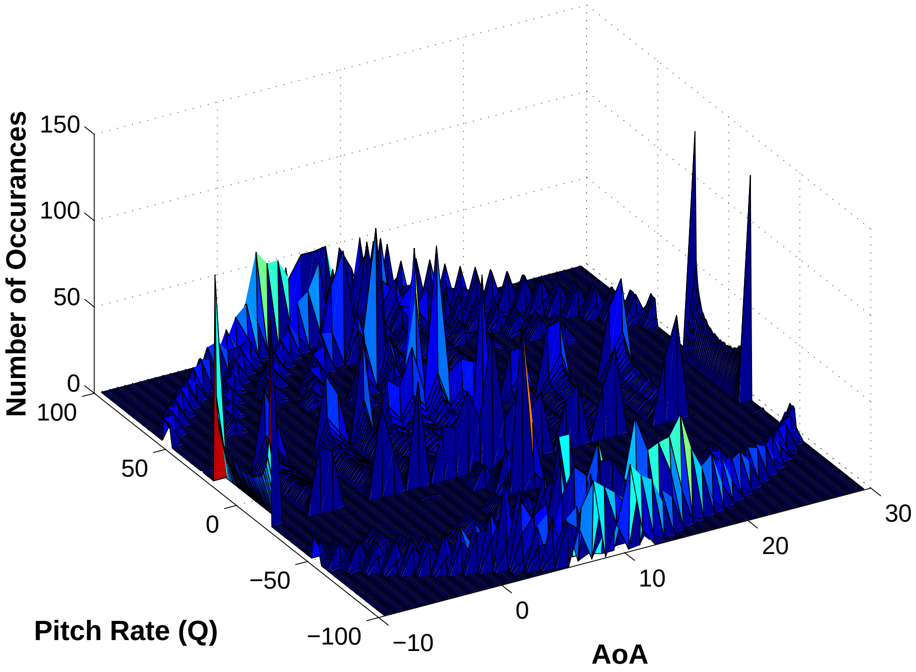


Figure 3.31: Discretized regressor space for sixth maneuver

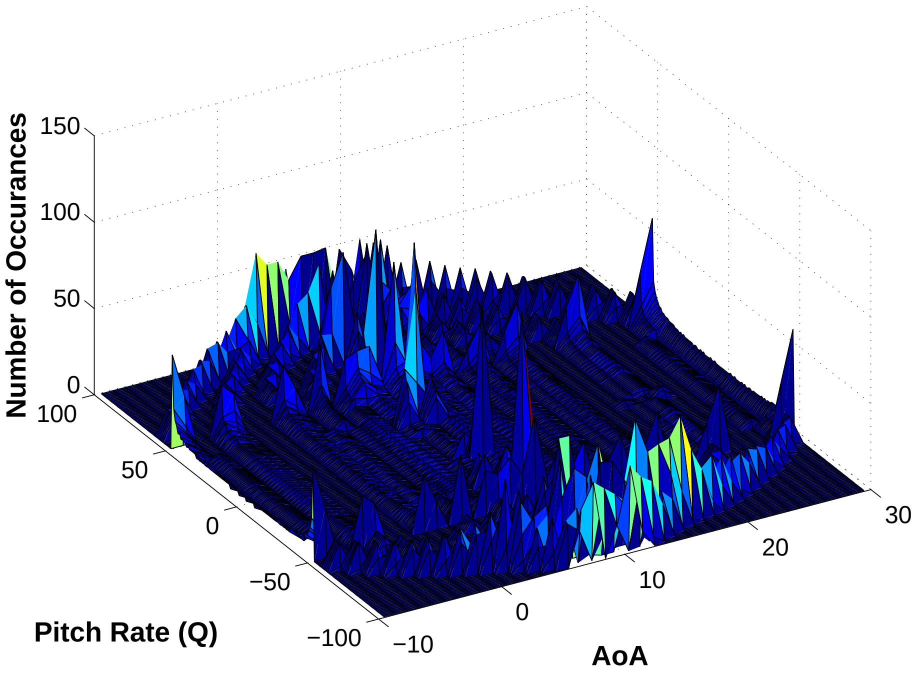


Figure 3.32: Discretized regressor space for seventh maneuver

cutoffs (Tables 3.3 - 3.6) again show very poor results in the coverage of both the low and high pitch rate regions.

From Table 3.2, the second maneuver (the current best) holds a significant lead in the overall coverage of the regressor space, a low coverage of the boundary cells, among the best (lowest) standard deviation in AoA, the best (lowest) standard deviation in pitch rate, and among the worst coverage in all zero pitch rate metrics. The other tables show very evenly spread data with average coverage of the low pitch rate regions and is consistently the best in all categories of high pitch rate (coverage, points per cell, and evenness). Maneuver 2 is superior to the first maneuver in nearly every metric. This provides some initial confirmation in the metrics as this should be the case for a set of valid metrics.

The third maneuver makes a significant sacrifice in the overall coverage of the regressor space and the boundary (Table 3.2) as well as the low and high pitch rate data (Table 3.3 through Table 3.6), in trade-off for the best number of points per cell and standard deviation of the zero pitch rate data.

Table 3.2: Output of first seven metrics for all seven maneuvers.

Maneuver	Percent		Std in		Data at 0 Q		
	Whole	Boundary	AoA	Q	Percent	Per Cell	Std
1	1.2730	8.0484	0.0688	0.1263	2.0000	0.0525	0.0606
2	2.4035	7.5063	0.0986	0.0789	2.5000	0.0275	0.0892
3	1.1100	1.2302	0.1644	0.1588	3.7500	25.3850	0.1136
4	1.4516	8.9241	0.1493	0.1557	6.5000	2.7275	0.0756
5	1.7545	11.3428	0.1249	0.1614	7.0000	0.2900	0.0672
6	1.9574	16.4095	0.1164	0.1910	4.2500	0.0500	0.1252
7	2.0841	16.6597	0.0700	0.1119	3.0000	0.0350	0.1047

Table 3.3: Output of last six metrics for 5 $\frac{deg}{s}$ cutoff between high and low Q.

Maneuver	Points Below 5 $\frac{deg}{s}$			Points Above 5 $\frac{deg}{s}$		
	Percent	Per Cell	std	Percent	Per Cell	Std
1	2.1350	0.0731	0.0670	1.2282	0.0286	0.0721
2	2.8775	0.0355	0.0907	2.3791	0.0305	0.1159
3	0.0725	0.0007	0.0130	1.1620	0.0190	0.0645
4	4.1200	0.1898	0.0825	1.3093	0.0210	0.0810
5	5.1625	0.1627	0.0771	1.5748	0.0237	0.0950
6	3.5075	0.0415	0.1185	1.8760	0.0302	0.0806
7	1.4375	0.0169	0.0855	2.1169	0.0315	0.0941

Table 3.4: Output of last six metrics for 10 $\frac{deg}{s}$ cutoff between high and low Q.

Maneuver	Points Below 10 $\frac{deg}{s}$			Points Above 10 $\frac{deg}{s}$		
	Percent	Per Cell	std	Percent	Per Cell	Std
1	2.0612	0.0696	0.0689	1.1859	0.0264	0.0721
2	2.7938	0.0346	0.1017	2.3605	0.0303	0.1161
3	0.0850	0.0013	0.0152	1.2214	0.0200	0.0672
4	2.8400	0.1071	0.0788	1.2939	0.0208	0.0813
5	4.2188	0.1077	0.0789	1.4788	0.0220	0.0959
6	3.5913	0.0438	0.1189	1.7771	0.0294	0.0785
7	1.4750	0.0177	0.0835	2.1509	0.0322	0.0948

Table 3.5: Output of last six metrics for 20 $\frac{deg}{s}$ cutoff between high and low Q.

Maneuver	Points Below 20 $\frac{deg}{s}$			Points Above 20 $\frac{deg}{s}$		
	Percent	Per Cell	std	Percent	Per Cell	Std
1	1.9600	0.0602	0.0719	1.1013	0.0234	0.0718
2	2.6956	0.0323	0.1135	2.3307	0.0303	0.1150
3	0.0831	0.0019	0.0145	1.3637	0.0221	0.0739
4	1.7638	0.0575	0.0763	1.3695	0.0224	0.0822
5	2.8950	0.0627	0.0957	1.4655	0.0226	0.0938
6	4.3719	0.0660	0.1215	1.3548	0.0225	0.0727
7	1.5131	0.0189	0.0834	2.2252	0.0337	0.0962

Table 3.6: Output of last six metrics for 50 $\frac{deg}{s}$ cutoff between high and low Q.

Maneuver	Points Below 50 $\frac{deg}{s}$			Points Above 50 $\frac{deg}{s}$		
	Percent	Per Cell	std	Percent	Per Cell	Std
1	1.6943	0.0436	0.0752	0.8519	0.0179	0.0685
2	2.5307	0.0289	0.1218	2.2755	0.0326	0.1075
3	0.2415	0.0044	0.0313	1.9743	0.0318	0.0928
4	1.1075	0.0280	0.0735	1.7885	0.0308	0.0885
5	1.8460	0.0324	0.0983	1.6563	0.0288	0.0900
6	1.9332	0.0297	0.0723	1.9770	0.0318	0.0926
7	2.2197	0.0302	0.0948	1.9520	0.0319	0.0925

Maneuvers 4 and 5 are similar in that they have average coverage of the regressor space and its boundary, have very good static data metrics (Maneuver 4 better than 5), and good to great coverage of the low pitch rate regions (5 better than 4 on average). The fourth maneuver has some poor results in the high pitch rate regions while the fifth maneuver is generally average in the same metrics. These two maneuvers are overall very similar as is expected with each excelling where it is expected as well.

The last two maneuvers have very good coverage of the entire regressor space and especially the boundary, but have average to poor zero pitch rate data. They vary from bad to good in low pitch rate data depending greatly on what the cutoff is. The sixth maneuver is good to poor in high pitch rate metrics while the seventh maneuver is excellent in the same.

3.4.3 Comparison Maneuvers. In order to validate the metrics and to determine which maneuver is best, a set of validation data must be available. Data used for this model comparison can be obtained from many outlets such as flight test, wind tunnel test, empirical results, and of course, CFD. This research will utilize the same CFD grid, solver, settings, etc. as the training maneuvers for the comparison maneuvers. When both the training maneuvers and comparison maneuvers are based on the CFD, any errors or discrepancies may be assumed to be modeling errors or CFD convergence issues instead of configuration or flight condition differences or any number of other possibilities [15].

When considering what types of comparison maneuvers to create, the purpose of the training maneuvers and validation considerations for the metrics should be taken into account. First, the model should be able to accurately predict a maneuver that covers as wide of a region in the regressor space as possible and so would be similar to the various training maneuvers. Second, the importance of the low pitch rate data has been a big focus so another maneuver must accurately cover this region. Last of all, the ability to predict static data is the primary downfall of the current “best” maneuver, so the last maneuver must focus on static regions in the regressor space.

The first comparison maneuver was created in a very similar manner to the training maneuvers. Equation (3.17) was employed to create this maneuver with the inputs given in Table 3.4.3. Toward the end of the maneuver, the bounds will exceed the -10 *deg* AoA limit of the regressor space so this value is chopped at -10 *deg* in the same manner of training Maneuvers 3 through 7. This, again, produces a sharp corner so this problem is solved by utilizing the spline function similar to training Maneuvers 4 and 5 except a sampling frequency of every $0.025/dt$ iterations so as to be dissimilar to both training maneuvers. This produces the maneuver seen in Figure 3.33 which covers the regressor space seen in Figure 3.34.

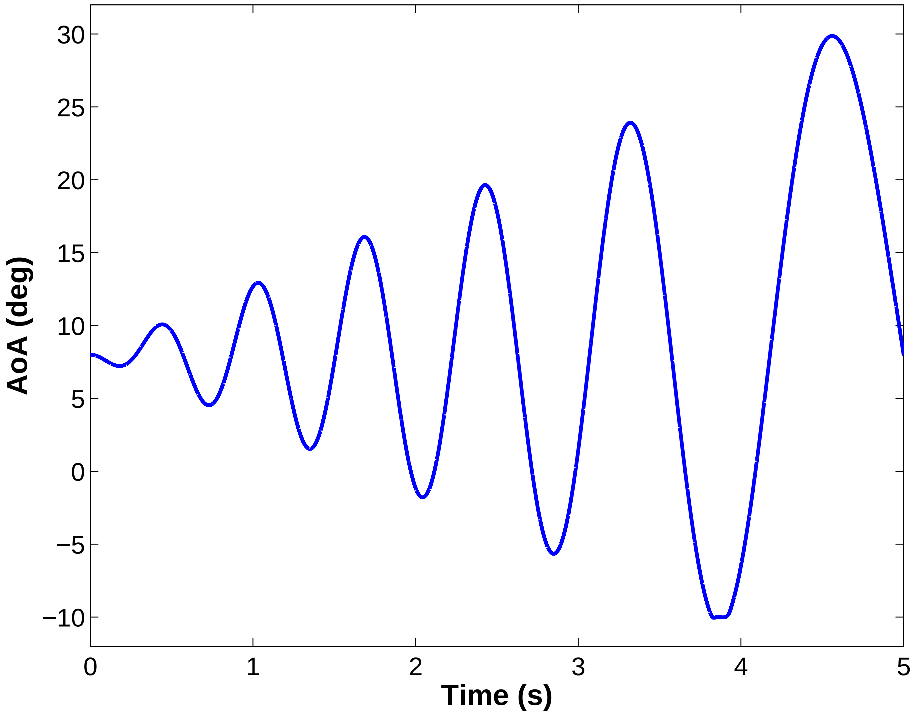


Figure 3.33: Motion of first comparison maneuver.

The second comparison maneuver focuses on low pitch rate entirely. For this purpose, the amplitudes of this motion are small. The inputs for Eq. (3.17) are again seen in Table 3.4.3. For this maneuver, however, the motion is only initially set for 2.5 s. The input values to Eq. (3.17) were established such that at 2.5 s, the maneuver can then be “mirrored” and produce an even transition from one to the next. This

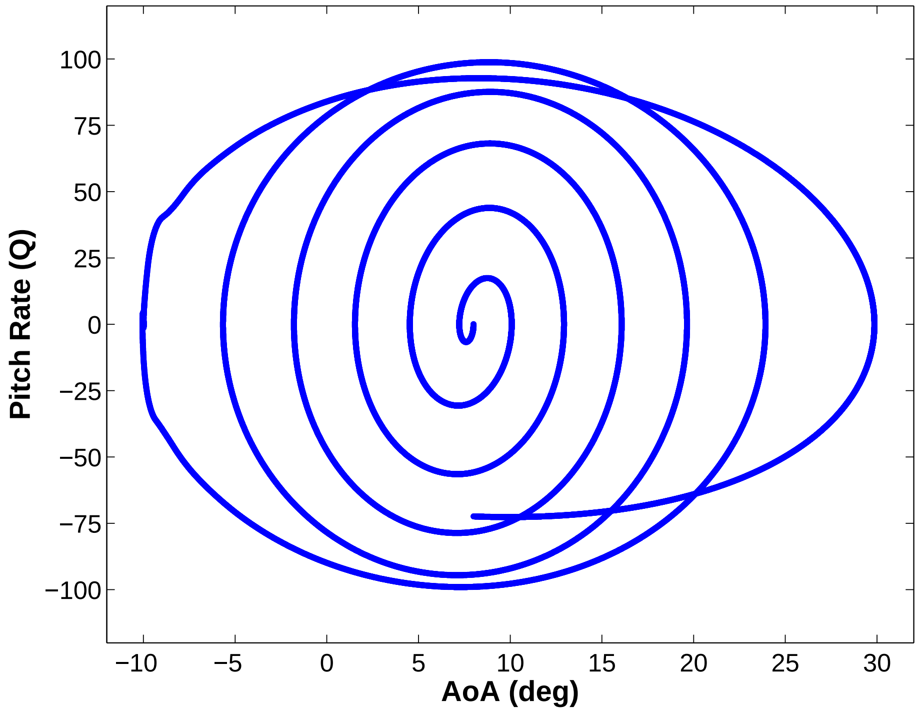


Figure 3.34: Regressor space covered by the first comparison maneuver.

Table 3.7: List of Inputs to Eq. (3.17) for each comparison maneuver.

Maneuver # (s)	AoA_i	a_1	a_2	f_1	f_2	λ_a	λ_f	Φ
1	8.0	0.0	24.0	1.8	1.2	1.0	1.2	90
2	8.0	1.0	4.0	0.6	0.8	1.0	1.0	-90

process produces the maneuver seen in Figure 3.35 with the evenly spaced regressor space seen in Figure 3.36. Notice the range of values on the axes of those two plots as they are smaller than every other maneuver shown thus far.

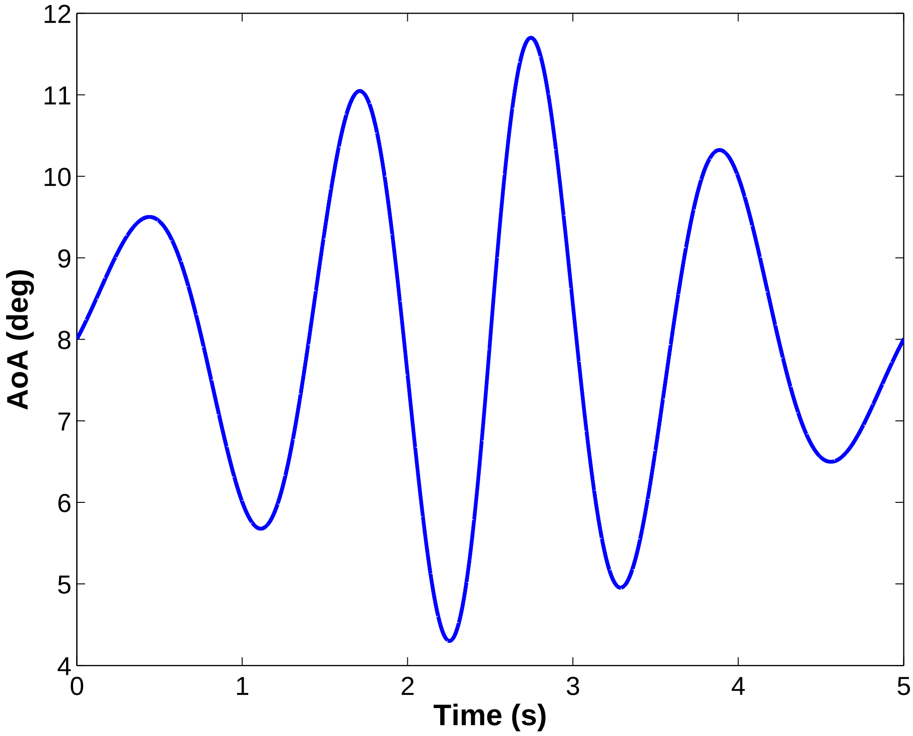


Figure 3.35: Motion of second comparison maneuver.

The last comparison maneuver for the static data was created directly in the Kestrel User Interface in the provided prescribed motion GUI feature. This feature includes simple sinusoidal motions as well as constant pitch rates an translations. For this last maneuver, the initial AoA will be at the bottom of the regressor space at -10 *deg* and will pitch up at a constant rate of 50 $\frac{deg}{s}$, holding at every 2.5 *deg* increments for 0.05 *seconds* (250 to 500 iterations) to allow the solution to converge to a static value, until the AoA reaches the top of the regressor at 30 *deg*. This stair-stepping approach can be seen in Figure 3.37 with its associated regressor space coverage, Figure 3.38.

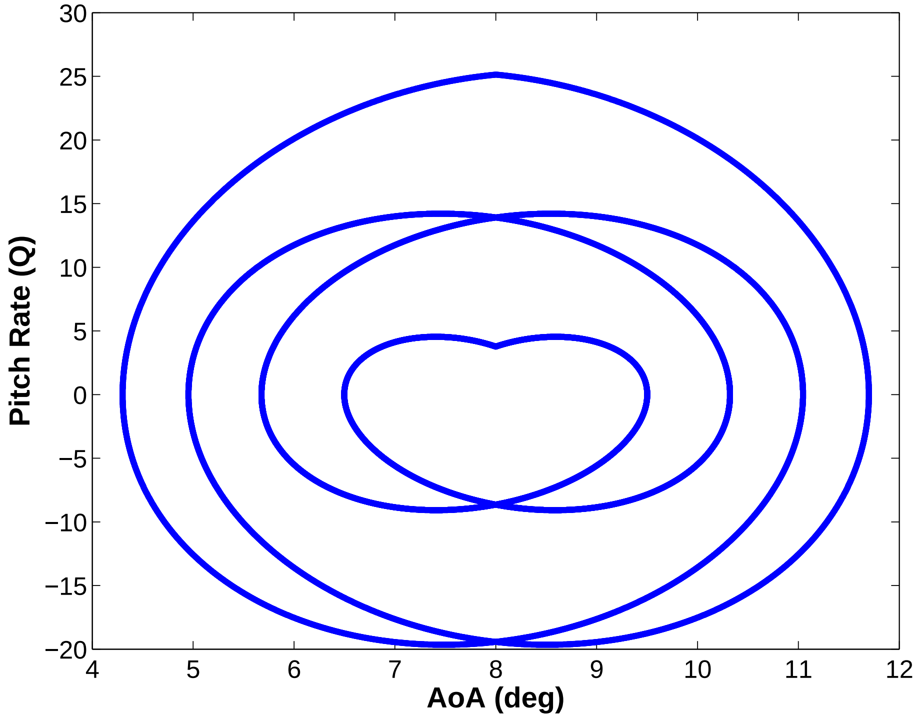


Figure 3.36: Regressor space covered by the second comparison maneuver.

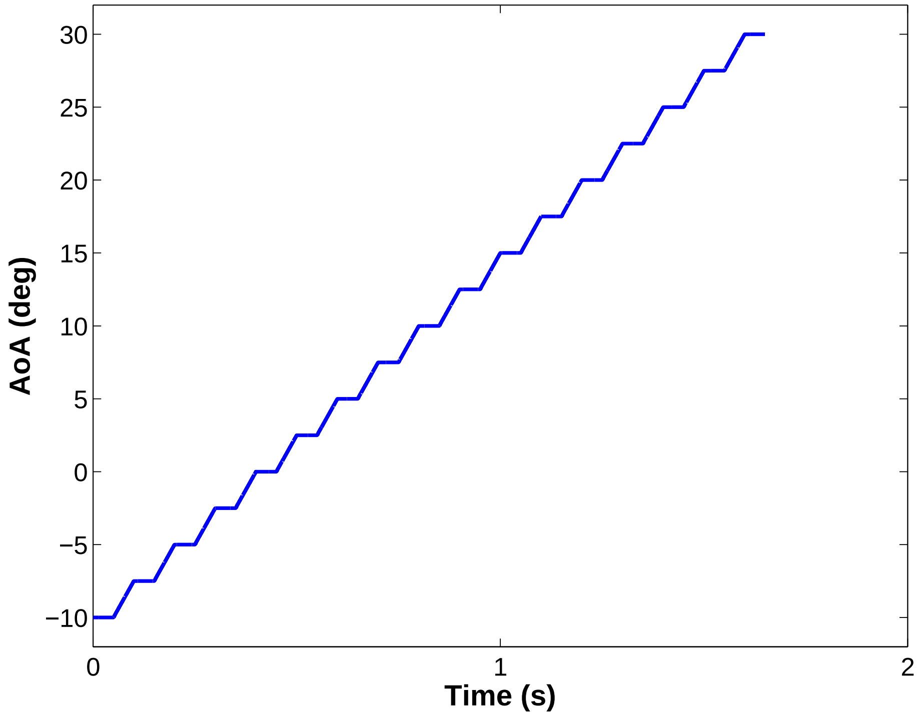


Figure 3.37: Motion of third comparison maneuver.

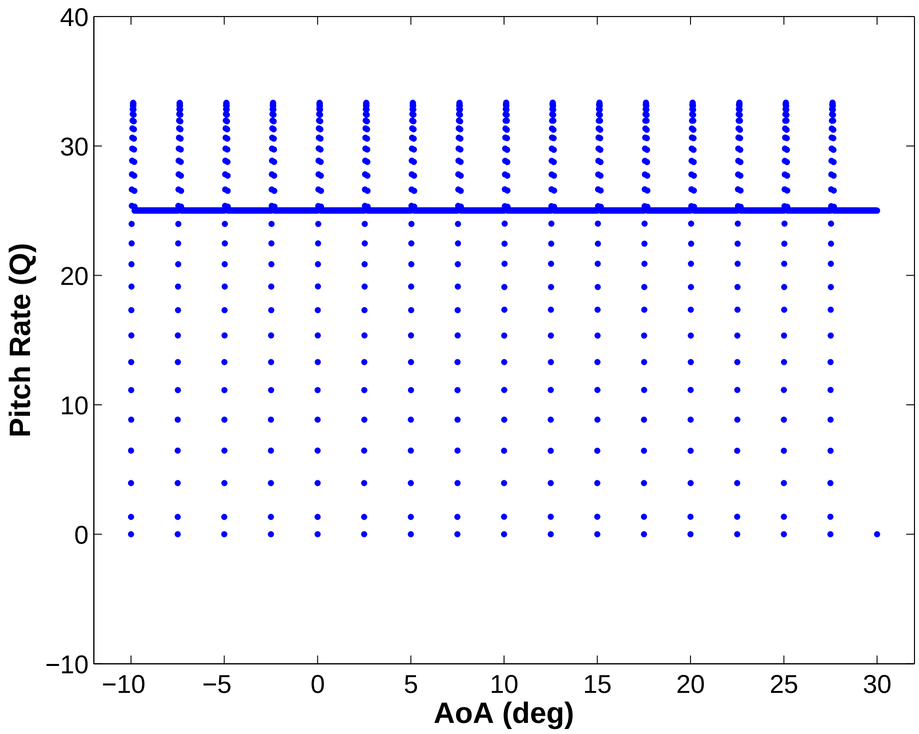


Figure 3.38: Regressor space covered by the third comparison maneuver.

3.4.4 Hypothesis. Maneuvers 6 and 7 are primarily used to give an example of no static data and a concentration of lower pitch rate data to hopefully highlight the importance of the static data and thereby, the static data metric. Conversely, Maneuver 3 has the most static data with very little low pitch rate data. All of these together should help to define a clear picture of the relative importance of each of these metrics.

The first maneuver should not be the best as the second maneuver is essentially an optimized form of the first as well as being the current best. The metrics also predict the first maneuver to be the worst. The second maneuver is able to introduce the most full, large-amplitude variations in AoA and therefore best overall coverage of the regressor space at a sacrifice of no static data.

It is hypothesized that either Maneuver 4 or 5 will ultimately provide the best match of the comparison sets of data as they provide the best blend between static and dynamic data with relatively even coverage between the two. These two motions were, on average, the median maneuvers according to the metrics which would seem to correlate a good blend between dynamic and static data.

It is also hypothesized that the metrics for the whole regressor space, zero pitch rate coverage (percentage and points per cell), a low pitch rate data of ± $5\frac{deg}{s}$ (percentage and points per cell), and the high pitch rate coverage of greater than ± $50\frac{deg}{s}$ will be the most important metrics so long as all maneuvers are evenly spaced well enough.

It is ultimately not important if these hypotheses are correct in the end as the results of which maneuver is the new (or remaining) best along with (potentially) validated metrics is the purpose of this research. It is sound scientific practice to set a theory and test it, however, and for that reason they have been stated herein.

3.5 Final Solver Settings

Section 2.2.1 discussed the mesh refinement process and its dependence on the solution of the flow field on the current grid. Therefore, before the grid refinement study can take place, the solver settings must be fully defined.

One of the biggest decisions to make is the choice of time step. Cummings et al. [13] develop a systematic process for properly performing a grid and time step study and how these studies can be performed in tandem to produce the best results. The grid size determines the size of flow phenomena that can be captured and the time step determines what frequencies of unsteadiness can be captured. Fortunately, however, the same paper gives accurate guidance for a proper time step for high AoA, delta wing type configurations and even examines a similar model of the F-16C in the results.

Cummings et al. [13] state in results and conclusions that a non-dimensional time step of $\Delta t^* = 0.01$ is probably adequate for obtaining physically realistic results in the leading edge vortex region and that this same value should be used as a starting point and their results show the correct time step is usually close to this value. The non-dimensional time step is given by:

$$\Delta t^* = \frac{\Delta t U_\infty}{l} \tag{3.39}$$

Where Δt is the dimensional time step, U_∞ is the freestream flow velocity, and l is a reference length for non-dimensionalization – often the mean aerodynamic chord. Since the examples used in this paper have similar (nearly identical for the F-16 case study) reference lengths, and the time step, solver settings, grid, etc., will all be identical for all maneuvers and comparison maneuvers, the time step for this research will follow this guideline and no time-step independence study will be performed. Also, any error in one maneuver or model will be inherent in all - canceling the effective error [15]. This does not, however, negate the premise of obtaining high-fidelity CFD results and the basis for the chosen time step is sound.

It is found that for the Mach 0.9 case, a time step of 0.0001 produces a non-dimensional time step of 0.008565 and for Mach 0.5 case, a time step of 0.0002 produces a non-dimensional time step of 0.00952. These values will then be set for all cases respective to each Mach number.

Another parameter of special importance is the number of Newton sub-iterations. Again, a study of the effect of this parameter on the solution is advised when performing CFD calculations to compare with flight test or another data set. For this research, a recommended practice given in the Kestrel user guide [12] of setting 3 to 5 sub-iterations for motion cases will be utilized. Due to the inherent unsteadiness in the expected flow characteristics and AoA values, the high end of 5 sub-iterations will be utilized for all cases and both Mach numbers.

One of the last parameters to mentioned in this section is the temporal damping values. It is common for CFD solvers to add numerical damping to the solved equations on one or both of the temporal and spacial set of terms. This non-physical damping promotes stability of the solver while damping out oscillations in the solution and hence altering the calculated solution. Since this damping is completely numerical, it is best to set these values to zero at least over time, but this is often infeasible.

For this purpose, the damping is decreased from 0.6 iteratively until the lowest value is found which produces a stable solution, giving the “best answer” for a given times step/sub-iteration combination on a certain grid. These three parameters are intermingled and by lowering time step, for example, the damping should be able to be decreased as well. These other values have already been set, however, and therefore the lowest value must be used for the calculations so long as the value is reasonable and it is common to all cases for a given Mach number. Kestrel only implements temporal damping on both the advection and diffusion terms. The Kestrel development team states that for the given release, damping values from 0.1 to 0.3 are reasonable and even up to 0.6 can be acceptable for motion cases [12].

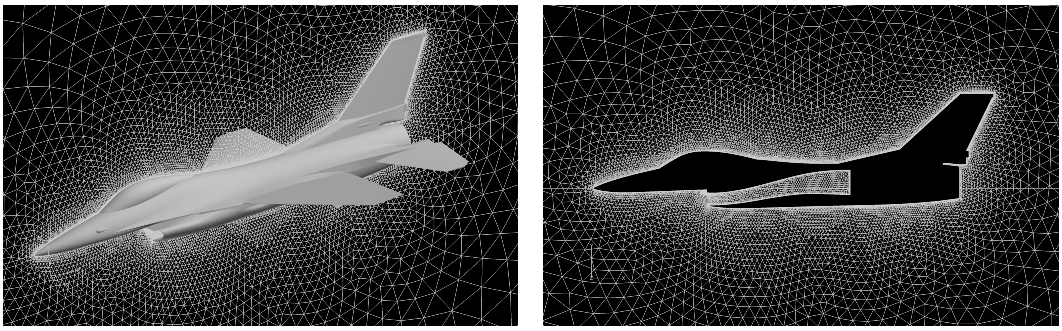
A complete list of solver settings can be found in Appendix B, but there are a few other settings to mention in this section. The solver will utilize the full unsteady N-S set of equations with the DDES turbulence model as mentioned in Section 2.1.6, with second-order temporal and spacial accurate solutions. Start-up iterations will also be utilized in which the boundary condition will be ramped from zero to the specified Mach numbers over 500 iterations and then steady state calculations will be calculated for a further 500 iterations before the time-accurate solution begins, starting with a temporal advection damping of 0.6 (and the default value for diffusion always being used of one-tenth the advection value), which is subsequently lowered with the use of restarts until the minimum value is reached.

3.5.1 Computational Resources. Performance of full aircraft CFD simulations requires a significant amount of computational power – more than is found in a normal research lab. For this purpose, the DOD Supercomputing Resource Center (SRC) will be utilized. In particular, the Raptor, Cray XE6 cluster will be utilized. It has login nodes populated with 2.7-GHz AMD Opteron 64-bit quad-core processors. The compute nodes have 2.4-GHz AMD Opteron 64-bit 8-core processors and there are 2732 of these compute nodes that share memory only on the node and not across nodes. Each compute node has two of the 8-core processors (16 cores) with its own Compute Node Linux operating system and 32 GBytes of DDR3 memory. Raptor's 1.6 PBytes of disk storage will also be utilized for the housing of large visualization and restart files [27].

3.6 Grid Refinement

It is noted that for the purpose of this research, the initial F-16C grid has been provided by the US Air Force SEEK EAGLE Office, which has been utilized in at least one paper produced by the office [15]. It is a full-scale model of the F-16 which is a slightly refined version of the grid used for the first direct-eddy simulation calculation over a full aircraft.

The model includes the forebody bump, diverter, and ventral fins (Figure 3.39(a)). The engine duct is also modeled and meshed up to the engine face (Figure 3.39(b)). The model is of a clean configuration with no wingtip missiles or attachment hardware and has an initial cell count of 6,847,832 cells with a refined wake region for high AoA studies. Since the grid is provided for this research, the grid generation was performed previously and details about the process are unknown, but the mesh has been utilized by authors of the grid and time step refinement paper [13] and has also



(a) Isometric view.

(b) Mesh up to engine face.

Figure 3.39: Views of the original grid.

been used to publish work in [15]. It cannot be assumed, however, that the grid was utilized for the flight conditions specified or the same range of AoA values. For this purpose, the grid refinement proposed in this section is accomplished.

It is also noted that it would be preferable to perform AMR on the given grid, but the AMR module implemented in Kestrel is not in complete working order and therefore a pseudo-AMR must be done. Instead, a different feature in Kestrel, where various types of geometric regions can be defined from which mesh refinement is accomplished, in a similar manner as AMR would have been used. The difference being that AMR defines the region by a vorticity iso-surface and the available capability refines a defined geometric region. These geometric regions will be defined such that they closely approximate the regions of highest vorticity which would be flagged for refinement in AMR. In this way similar results may be gained (much better vortex refinement without drastic increases in mesh size).

The process of refining by a region is much more time-consuming and it is infeasible to examine the desired Q-threshold values over the length of an entire maneuver (even with AMR the computational cost is increased significantly so as so dismiss such a process for the scope of this work). Therefore, only the extreme values of AoA will be refined and only one level of refinement will be done.

In other words, the solution of the original grid will be compared to other solutions based on different defined regions (of varied size) to understand the change in the solution answer from one grid to the next. Ideally, the refined grids would be again refined based on regions of the highest Q-threshold again to see how much the solution changes in that iteration of refinement. Even though refining based on a region saves on computational cost by decreasing cell count when compared to refining the entire mesh, the computational cost will still rise quickly.

Grid convergence for this work is not as important as if the comparison data came from another source such as flight test, since the comparison maneuvers will be run on the same grid as the rest of the maneuvers and the spacial resolution will be the same for all cases and any error resulting from this will fall out of the problem. The one level of grid refinement will still be performed at both 30 *deg* and −10 *deg* AoA as high-fidelity CFD simulations are still desired. Performing this “one level” of grid refinement may not produce a grid converged solution, but will at least help to quantify the type of error seen from the original grid the first levels of refinement.

The last note for the provided mesh is the boundary conditions. This grid includes adiabatic solid wall for the surface of the aircraft and the engine duct, and modified Riemann invariants for the far-field boundaries. Also, a source boundary condition is used to create an inflow condition at the engine exhaust (based on Riemann invariants) and a sink boundary condition is used at the engine face to model the corrected engine mass flow. The values for these can be found in Appendix B.

3.6.1 High AoA Refinement. The first CFD simulation is done using the original grid and the solver settings mentioned in Section 3.5 and Appendix B. This simulation will provide the baseline results for each Mach number at the 30 *deg* AoA level and also provide the basis for defining regions for grid refinement for comparing solution changes between grids.

After the first solutions were calculated, it was determined to examine two levels of Q-threshold from which to define regions for refinement. Figure 3.40 shows the results for Mach 0.5 at the stated AoA level and a Q-threshold value of 1.0, colored by vorticity magnitude. Similarly, Figure 3.41 displays similar results for the Mach 0.9 case. Those figures show very large regions and is difficult to see where the regions of highest vorticity are located. For this reason, another view is given with a Q-threshold value of 2.5 as is seen for the Mach 0.5 case (Figure 3.42) and the Mach 0.9 case (Figure 3.43).

Figure 3.42 and Figure 3.43 show very similar regions where the highest vorticity lies due to the strong vortex core created by the strakes, with the Mach 0.5 case covering a slightly larger region. This flow field is highly separated in all regions - including the vortex core from the strake. This region of highest unsteady flow from

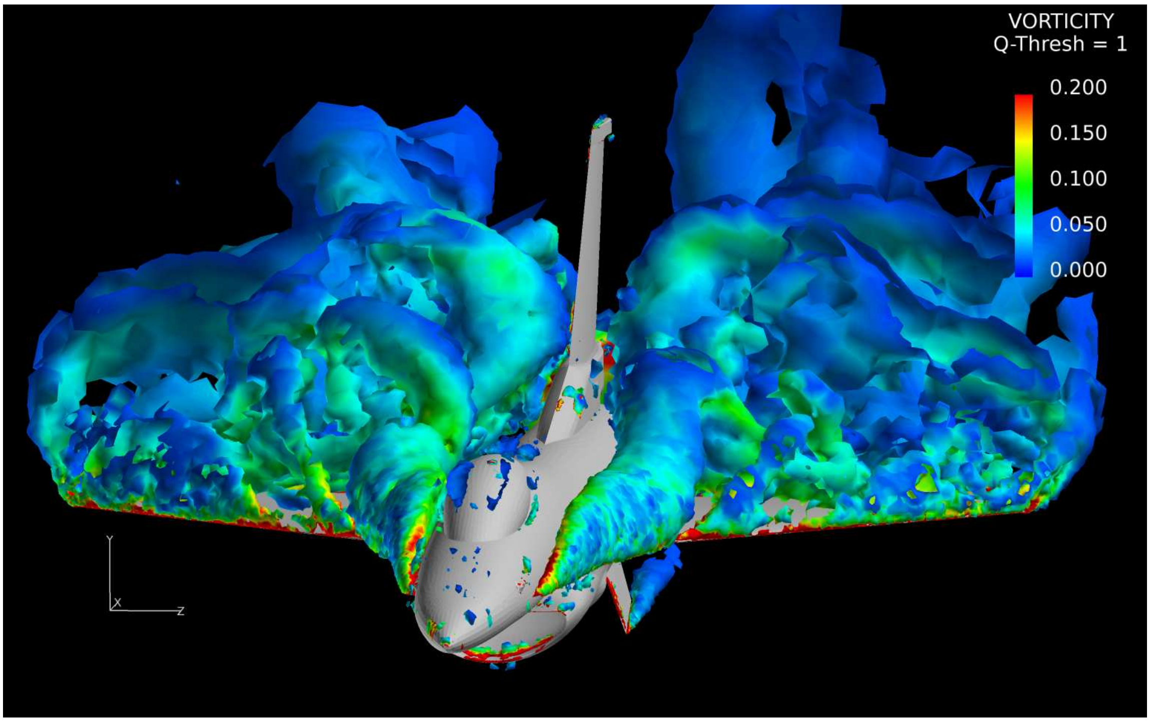


Figure 3.40: Flow solution of initial grid at Mach 0.5 and AoA = 30.

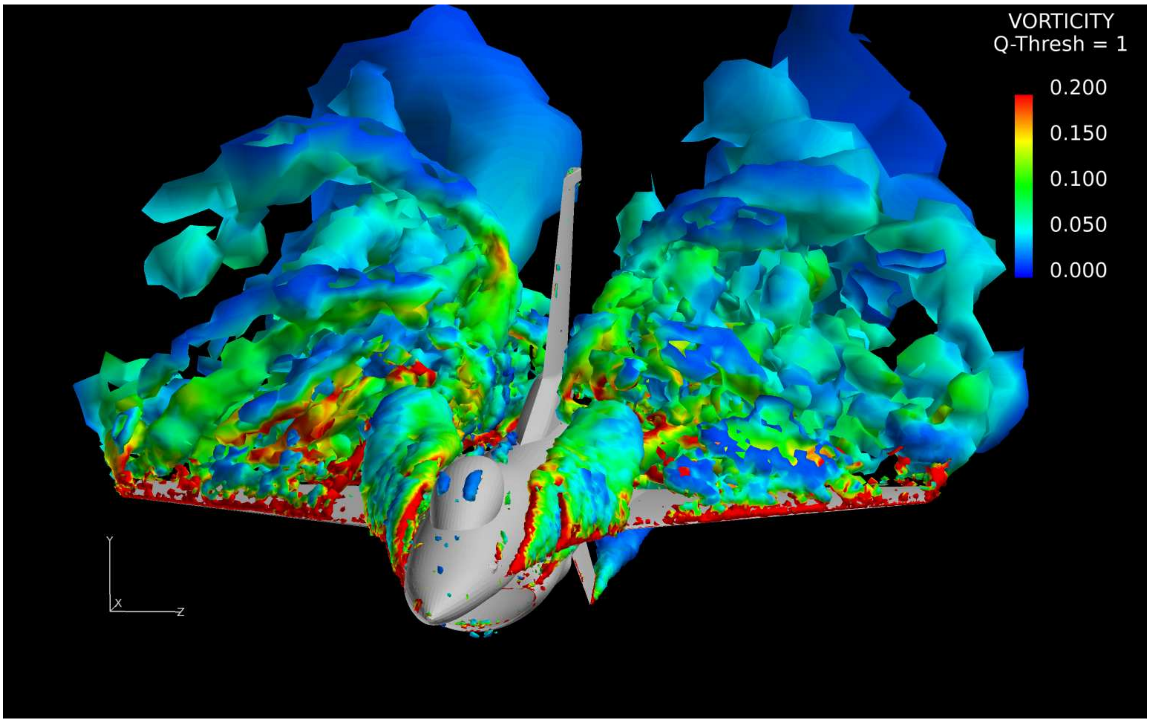


Figure 3.41: Flow solution of initial grid at Mach 0.9 and AoA = 30.

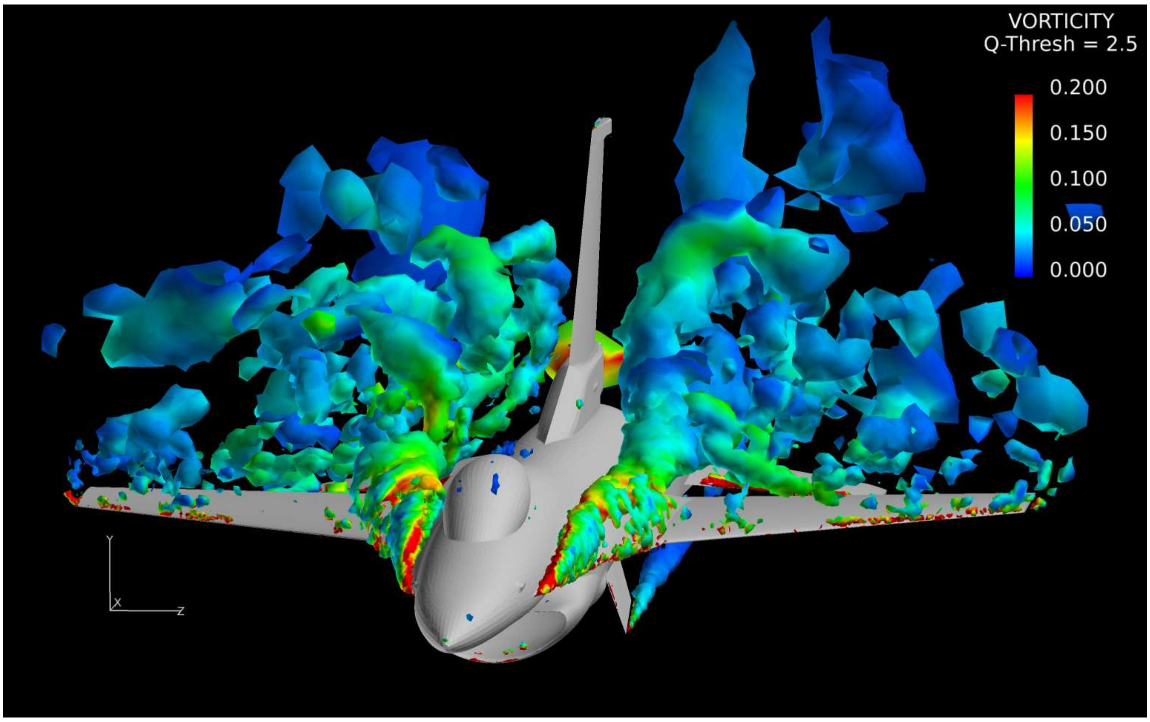


Figure 3.42: Flow solution of initial grid at Mach 0.5 and AoA = 30.

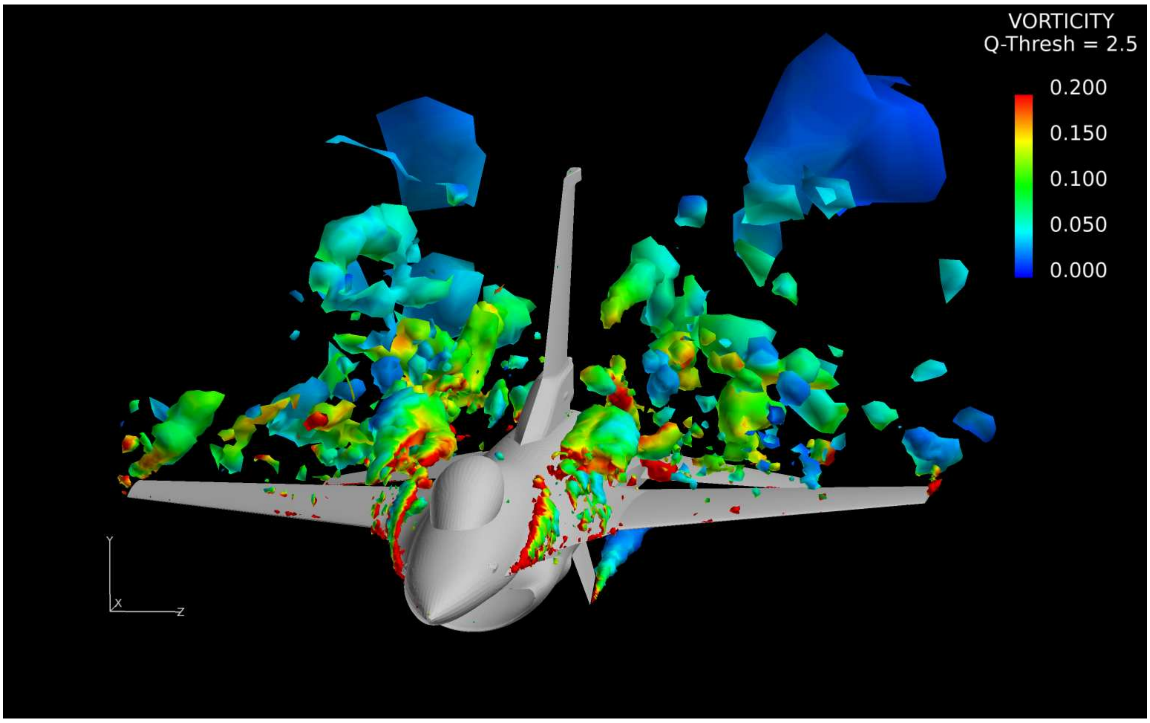


Figure 3.43: Flow solution of initial grid at Mach 0.9 and AoA = 30.

the strake will be the region of interest for the first grid refinement. This region is defined by a cone starting above the strake and going backwards from there, increasing in size to cover the area of the vortex core on each side of the aircraft. This grid is seen in Figure 3.44, Figure 3.45, and Figure 3.46. This second grid roughly doubles the cell count from the original grid to a total of 13,403,970 cells.

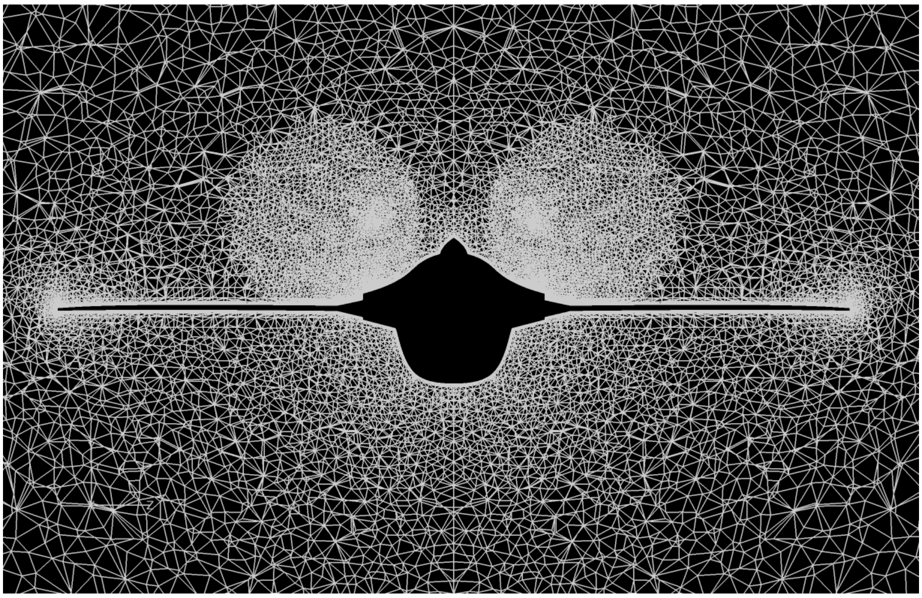


Figure 3.44: First refined region for AoA = 30 *deg* in X direction.

The next region defined for this case is much larger and covers nearly the entire area seen in Figure 3.40 and Figure 3.41. This region is defined by two hexahedron, one on each side of the symmetry plane of the aircraft. For the purpose of showing the refined region, Figure 3.47, Figure 3.48, and Figure 3.49 each display the intermediate step where only one side of the symmetry plane had been refined. This produces the original grid on one half and the newly refined region on the other. This third grid is almost four times larger than the original grid at 23,246,440 cells.

Two new grids have now been created based off of the results in the original grid. Solutions are now calculated using these two new grids and the coefficients of interest are plotted over time so the variation of these values for the three grids can be seen. These results show good grid convergence between the second and third

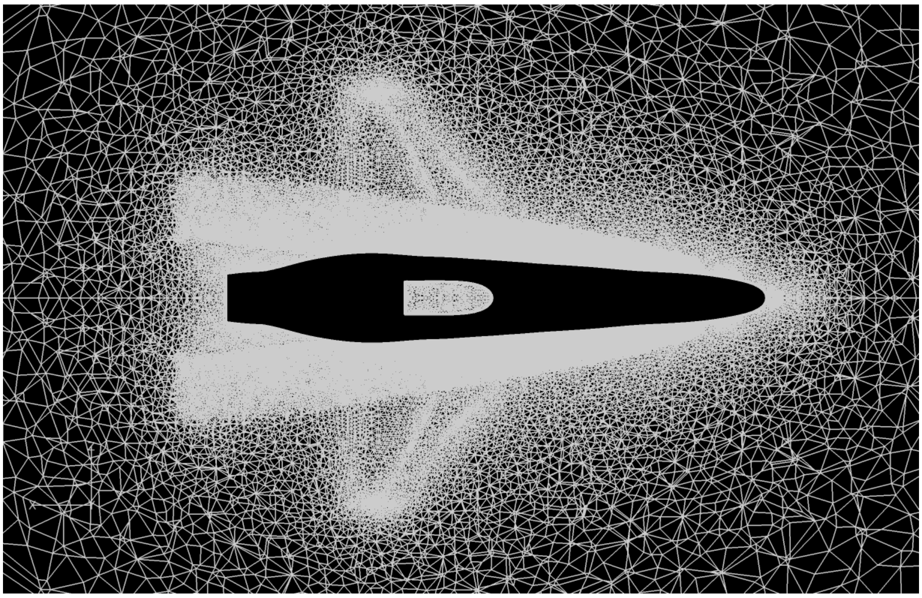


Figure 3.45: First refined region for AoA = 30 *deg* in Y direction.

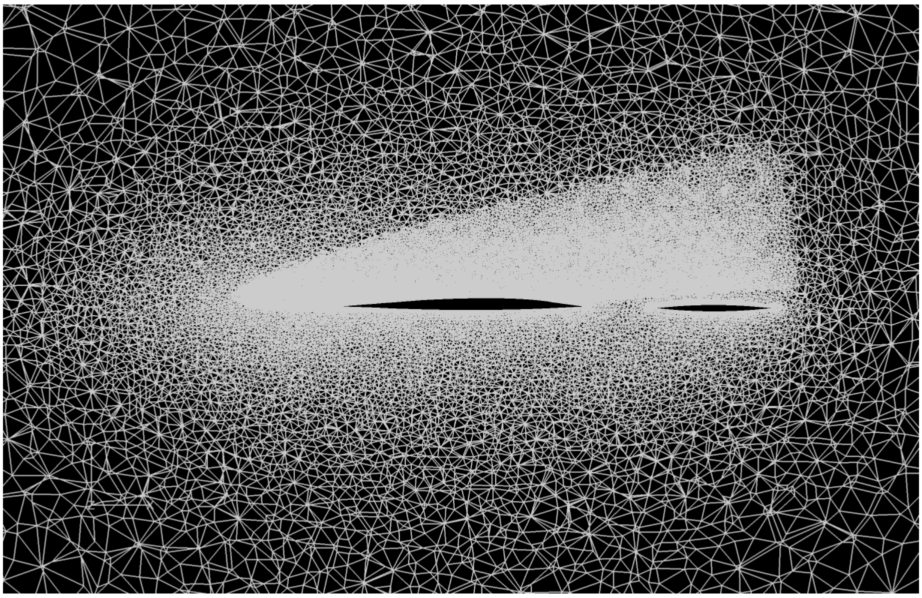


Figure 3.46: First refined region for AoA = 30 *deg* in Z direction.

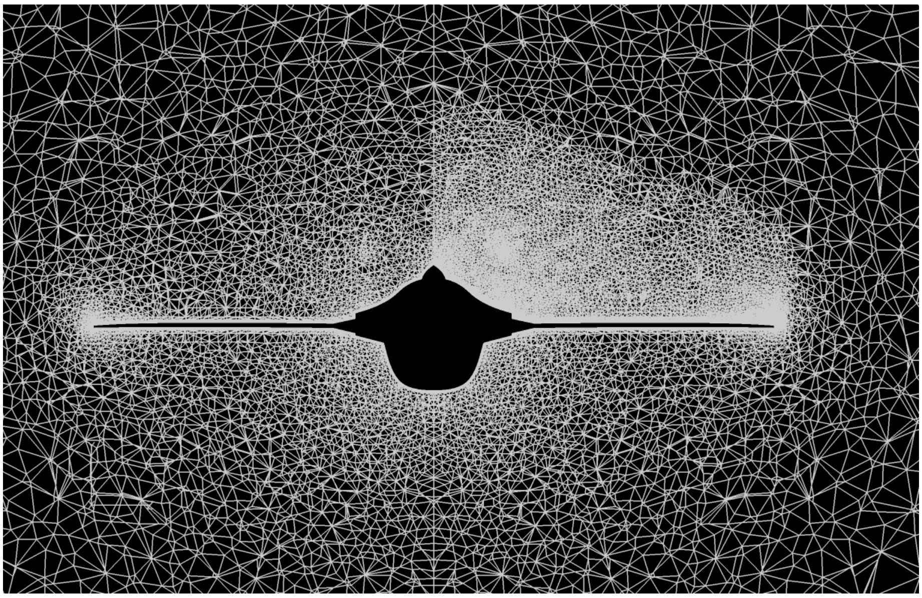


Figure 3.47: Second refined region for AoA = 30 deg in X direction.

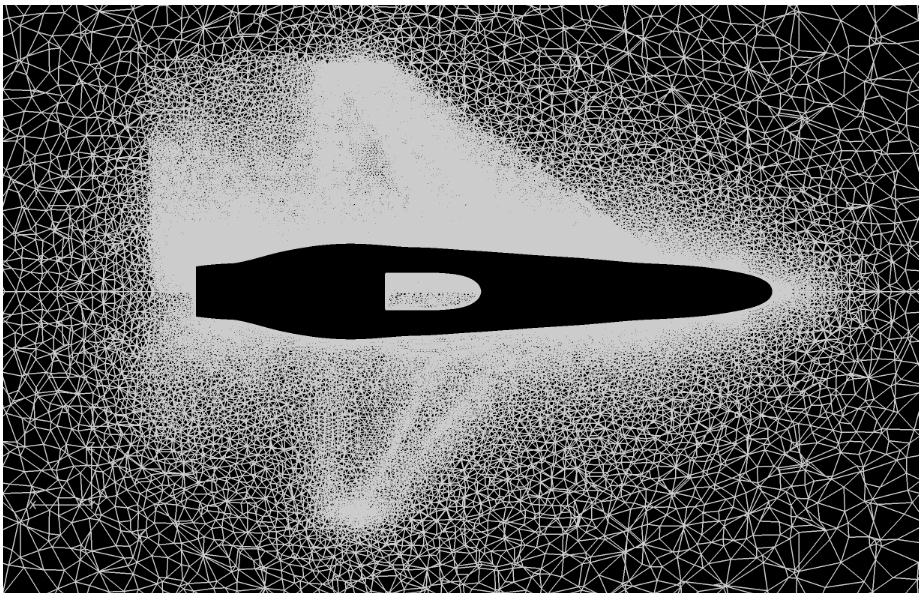


Figure 3.48: Second refined region for AoA = 30 deg in Y direction.

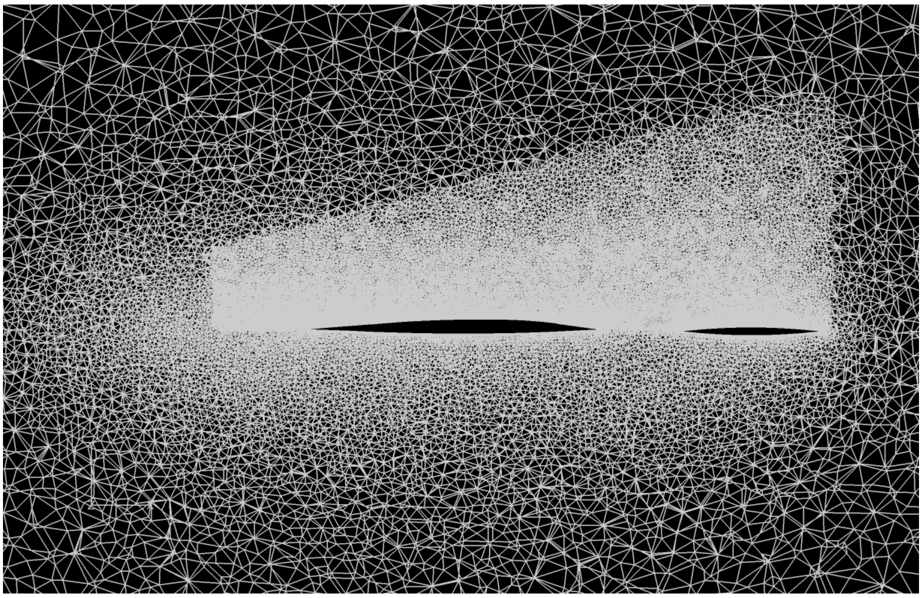
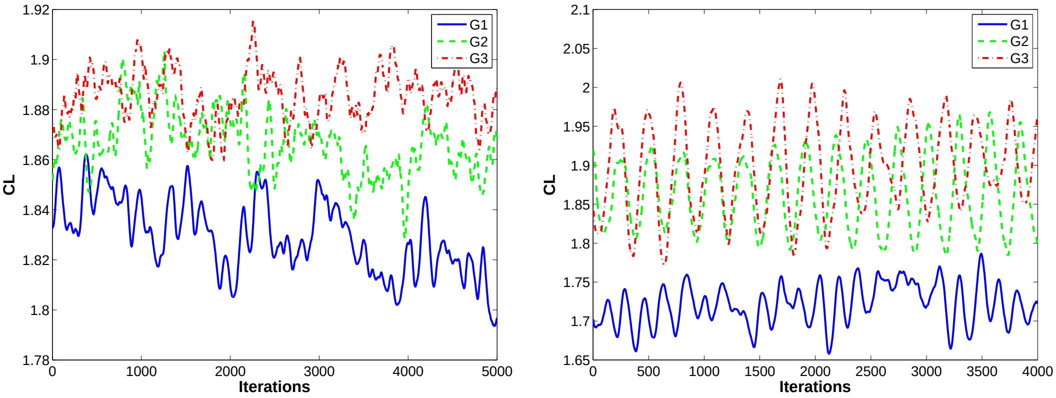


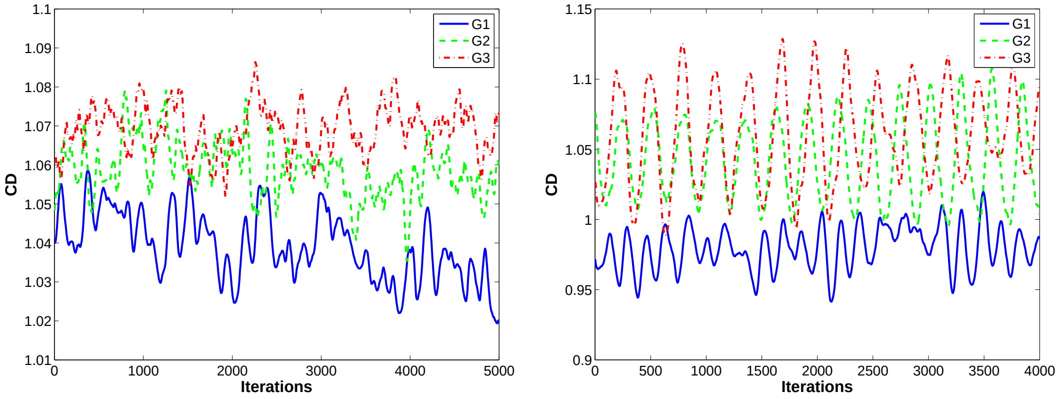
Figure 3.49: Second refined region for AoA = 30 *deg* in Z direction.



(a) Mach = 0.5

(b) Mach = 0.9

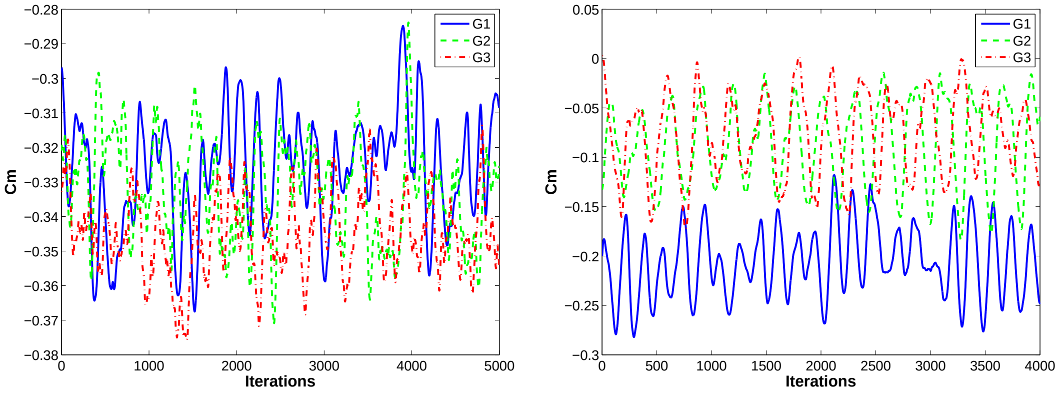
Figure 3.50: Grid refinement results for lift at 30 *deg* AoA.



(a) Mach = 0.5

(b) Mach = 0.9

Figure 3.51: Grid refinement results for drag at 30 *deg* AoA.



(a) Mach = 0.5

(b) Mach = 0.9

Figure 3.52: Grid refinement results for pitching moment at 30 *deg* AoA.

grid for both Mach numbers. The mean values and relative size of oscillations of the coefficients are very similar and the change in mean value from the first grid to the other two is relatively small considering the relatively small range on the y-axes of the plots. Since the second and third grids are very similar in their answers, the second grid is picked as the proper level of refinement with fewer cells to proceed forward to grid refinement at the low AoA.

3.6.2 Low AoA Refinement. The "initial" grid for this -10 *deg* AoA refinement is the second grid from the high AoA refinement. The initial CFD simulation is again calculated on Grid 2 from above and can be examined to help determine which regions should be refined for this case. The same two levels of Q-threshold from the high AoA will again be examined. Figure 3.53 shows the results for Mach 0.5 at the stated AoA level and a Q-threshold value of 1.0, colored by vorticity magnitude. Similarly, Figure 3.54 displays similar results for the Mach 0.9 case. Those figures show much smaller regions than the corresponding plots from the high AoA case. There

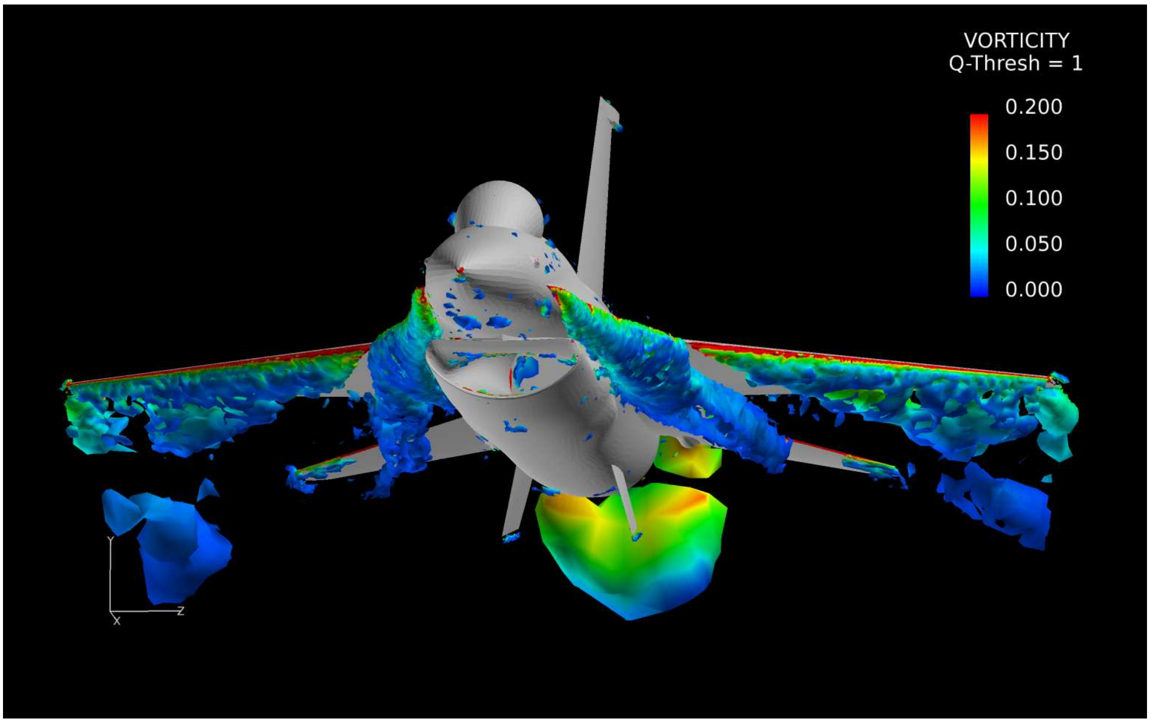


Figure 3.53: Flow solution of grid at Mach 0.5 and AoA = -10.

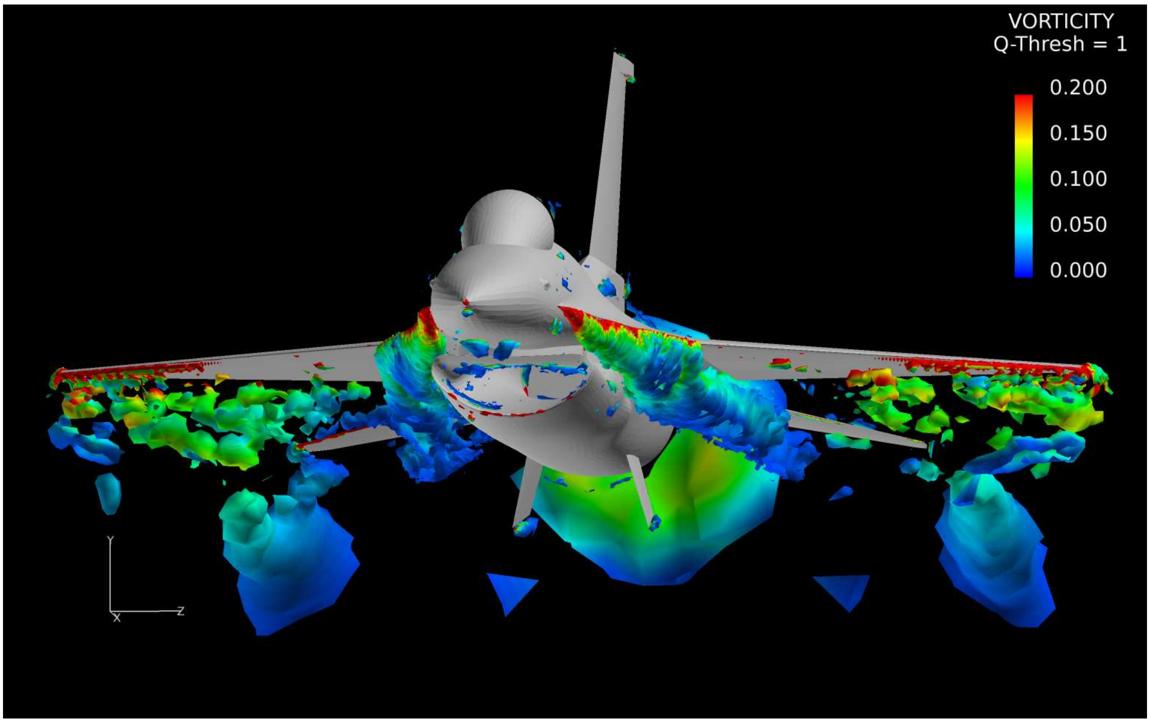


Figure 3.54: Flow solution of grid at Mach 0.9 and AoA = -10.

is a well defined vortex core from the strakes again although this time they appear to stay attached. Outbound on the wing, there is some vorticity which is off of the surface and although it stays fairly immovable on this grid over time, it may become unsteady with further refinement.

Looking at the other Q-threshold value of 2.5 again, again shows a strong vortex core and little else as is seen in both Mach numbers (Figure 3.55 and Figure 3.55).

The second grid for this -10 *deg* AoA case will be again attempt to further refine the vortex core as this is the source of highest overall vorticity. This region is again defined by a cone starting below the strake and going backwards from there, increasing in size to cover the area of the vortex core on each side of the aircraft. This grid is seen in Figure 3.57, Figure 3.58, and Figure 3.59. This second grid adds roughly 1.4 million cells to a total of 14,802,078 cells.

For this AoA, an extra grid will be utilized in which the outboard regions under the wing will be refined and the vortex core region will *not* be refined for this third

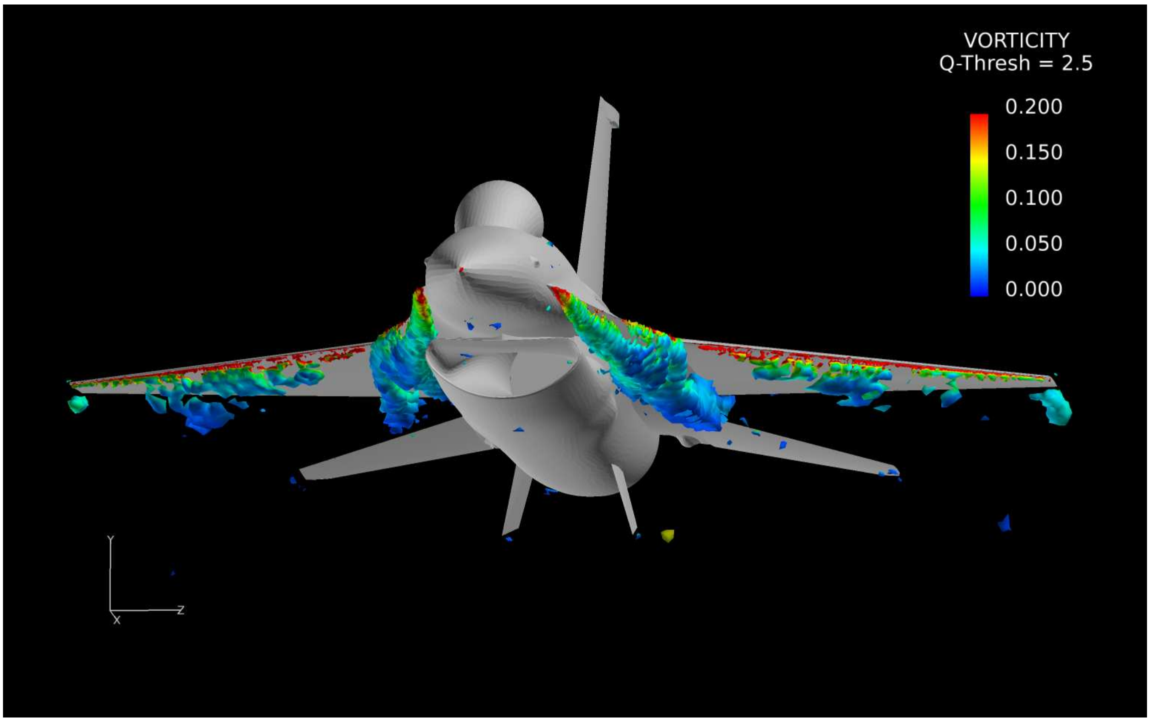


Figure 3.55: Flow solution of grid at Mach 0.5 and AoA = -10.

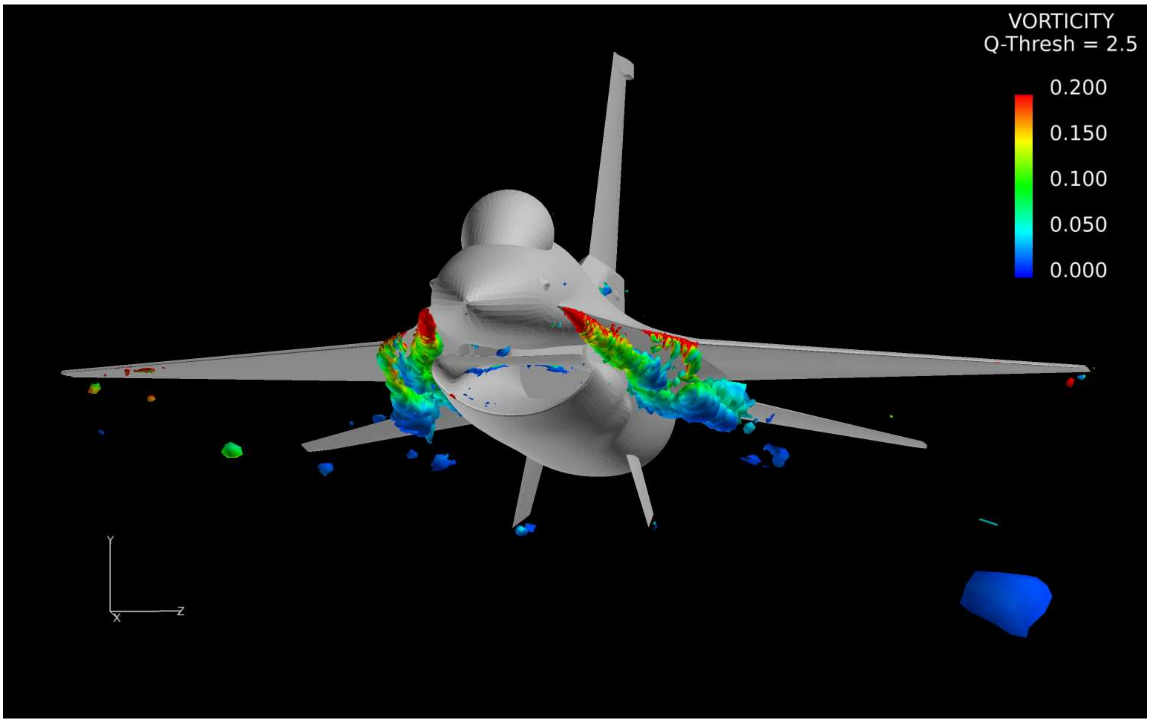


Figure 3.56: Flow solution of grid at Mach 0.9 and AoA = -10.

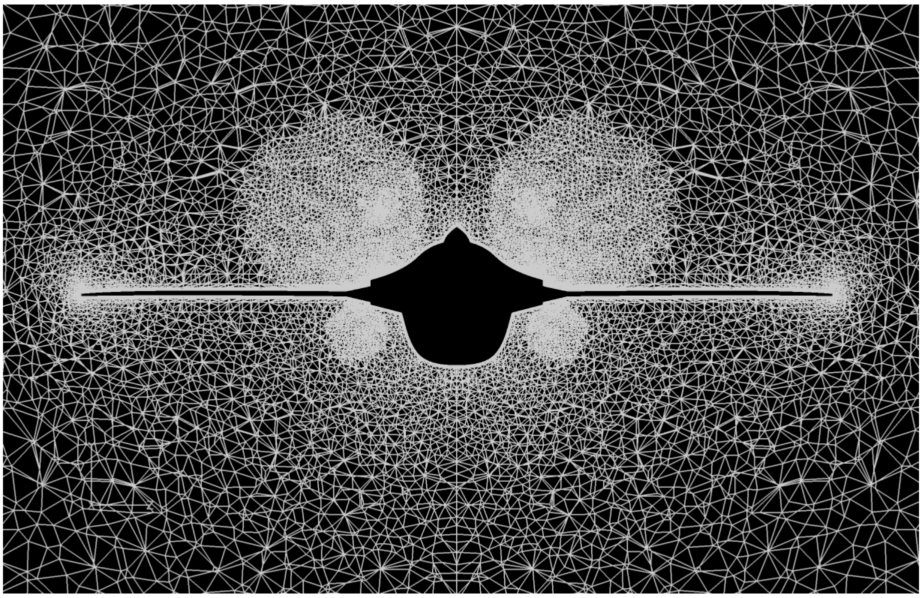


Figure 3.57: First refined region for AoA = −10 *deg* in X direction.

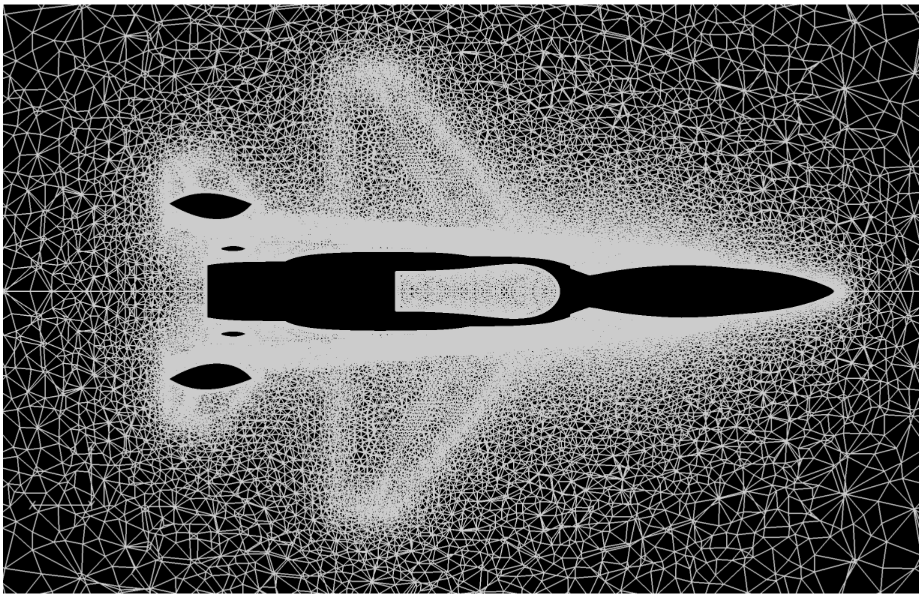


Figure 3.58: First refined region for AoA = −10 *deg* in Y direction.

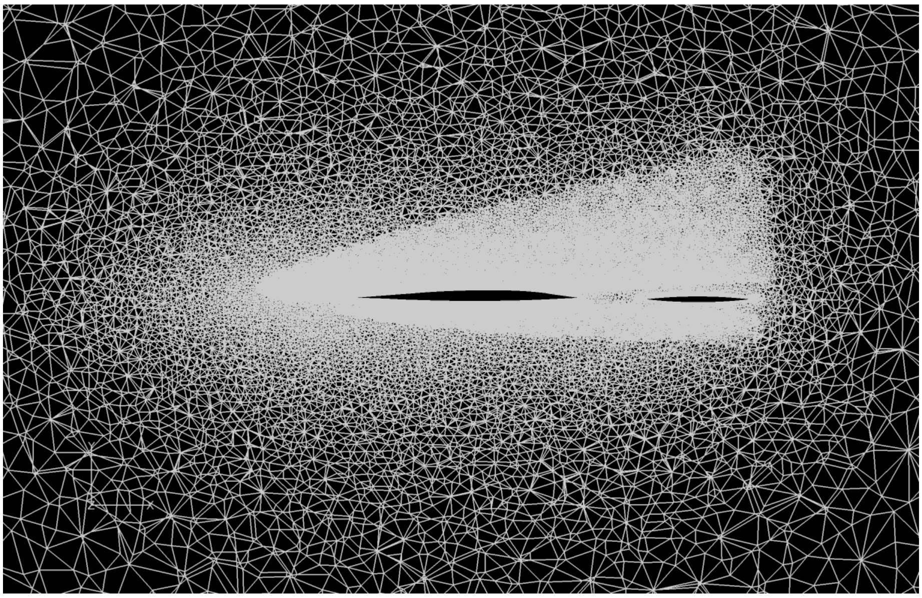


Figure 3.59: First refined region for AoA = -10 *deg* in Z direction.

grid. This is due to the vortical structures seen outboard of the wing which although steady, appear as though they may become unsteady with further refinement. Grid 3 for this AoA is seen in Figure 3.60, Figure 3.61, and Figure 3.62. This grid region is defined by a hexahedron and totals roughly the same number of cells as Grid 2, numbering at 14,341,214 cells.

The third and last region defined for this case is similar to the last grid for the 30 *deg* AoA case. It is much larger than the previous two and covers nearly the entire area seen in Figure 3.53 and Figure 3.53. This region is again defined by two hexahedron, one on each side of the symmetry plane of the aircraft. Similar to the final grid for the high AoA, only half of this refined mesh is shown for the purpose of showing the distinction of the refined region versus the original (Figure 3.47, Figure 3.48, and Figure 3.49). This fourth grid is almost twice as big as the original grid for this AoA, numbering at 26,509,412 cells.

There are now four grids available for this AoA. Each is used in CFD calculations with the exact same settings as stated in Section 3.5 and Appendix B. The lift, drag, and pitching moment coefficients for each grid is then samples over thousands of

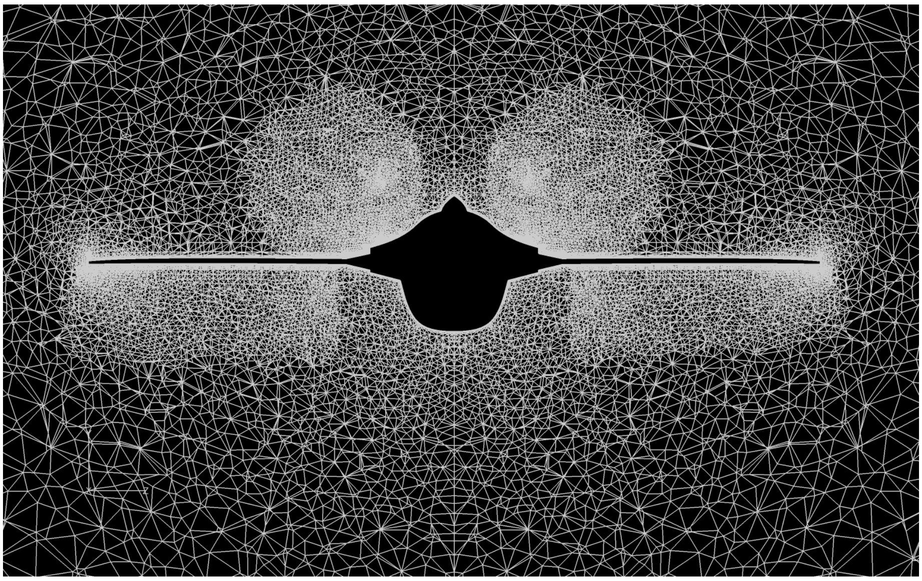


Figure 3.60: Second refined region for AoA = -10 *deg* in X direction.

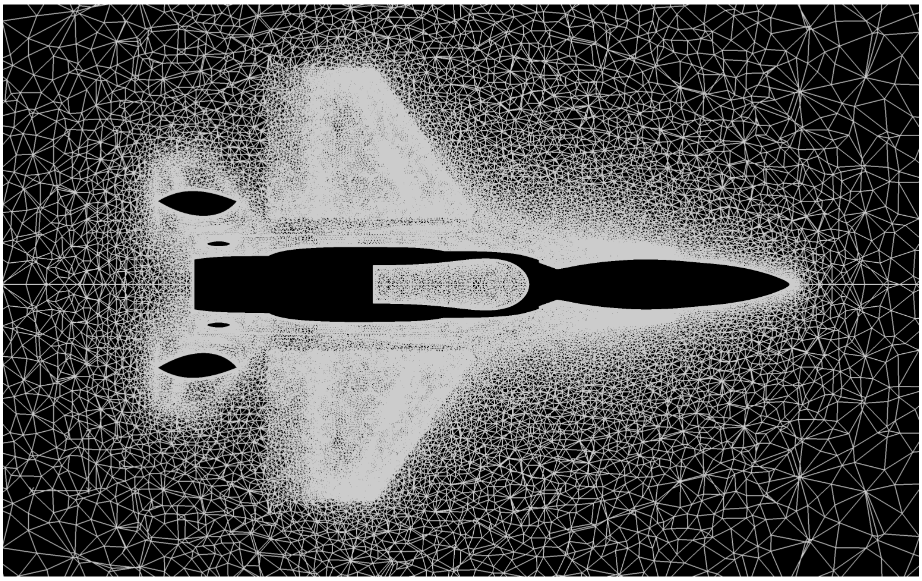


Figure 3.61: Second refined region for AoA = -10 *deg* in Y direction.

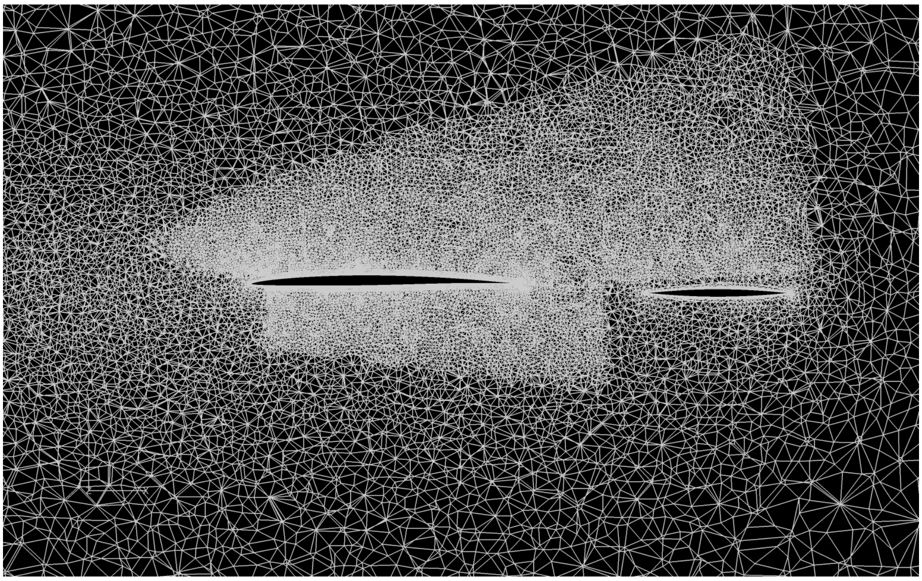


Figure 3.62: Second refined region for AoA $= -10$ *deg* in Z direction.

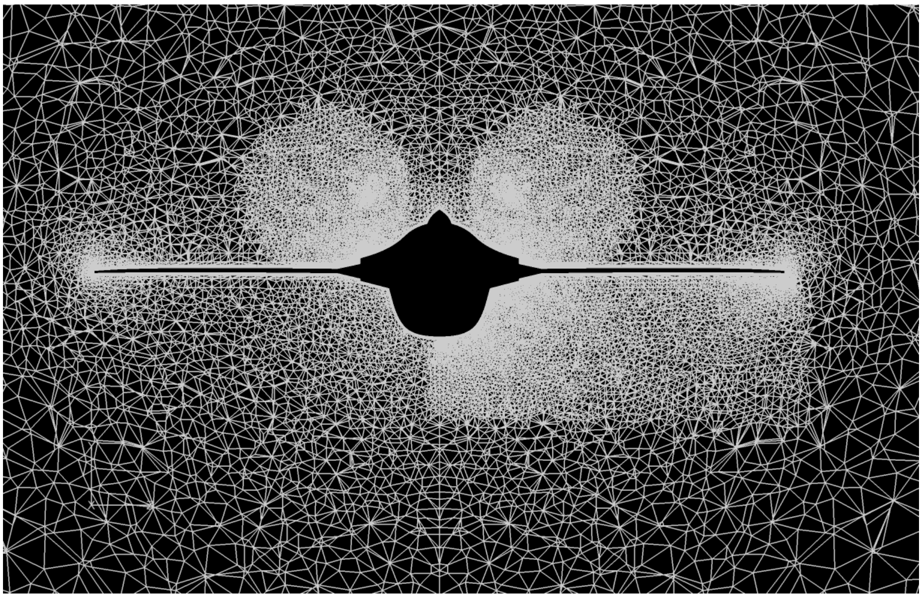


Figure 3.63: Third refined region for AoA $= -10$ *deg* in X direction.

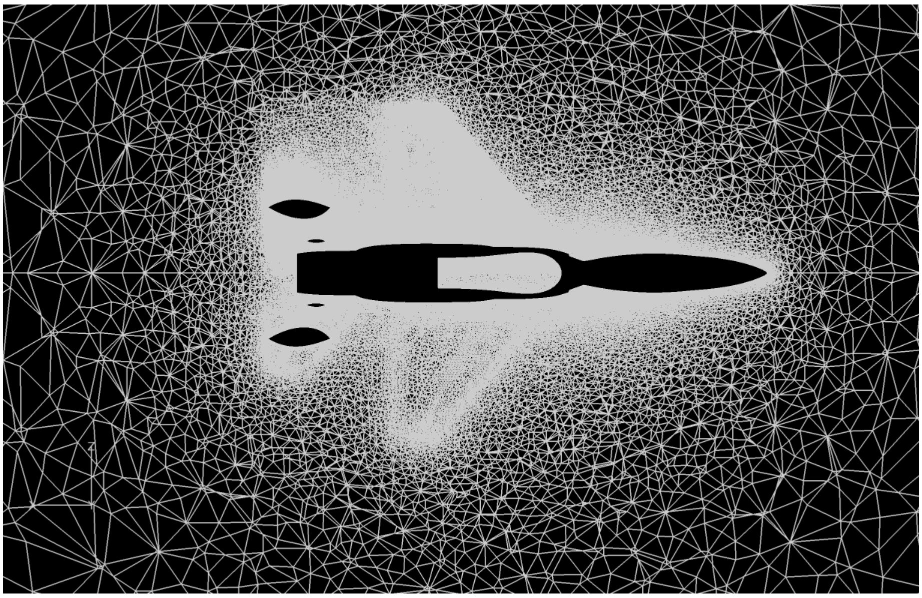


Figure 3.64: Third refined region for AoA = -10 *deg* in Y direction.

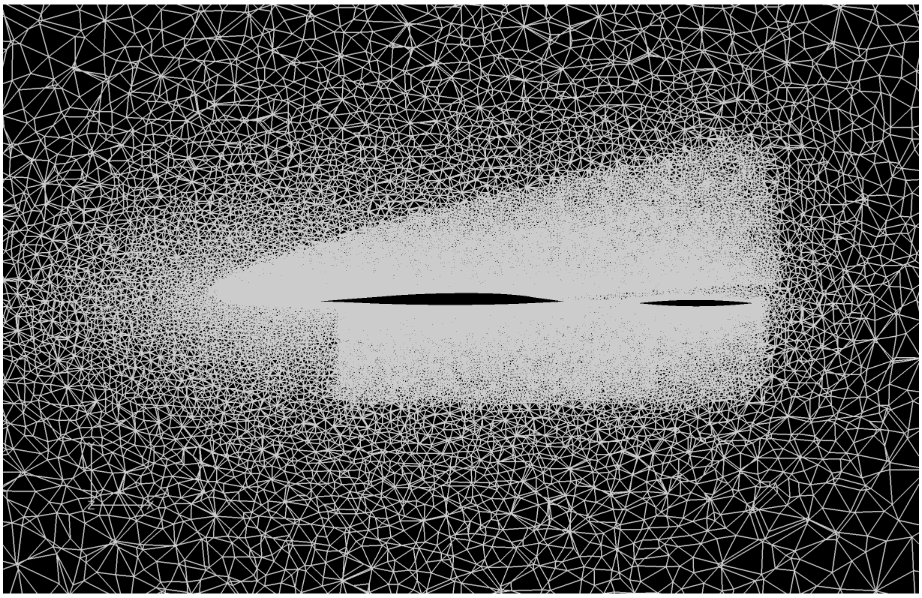
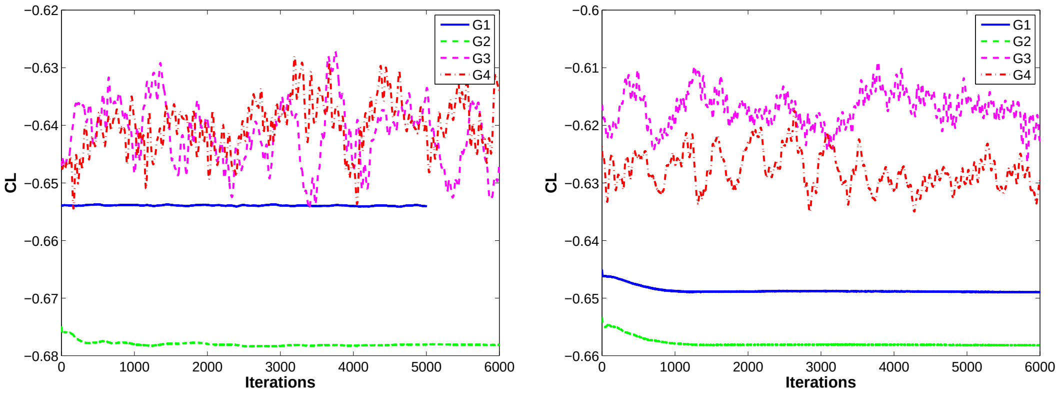


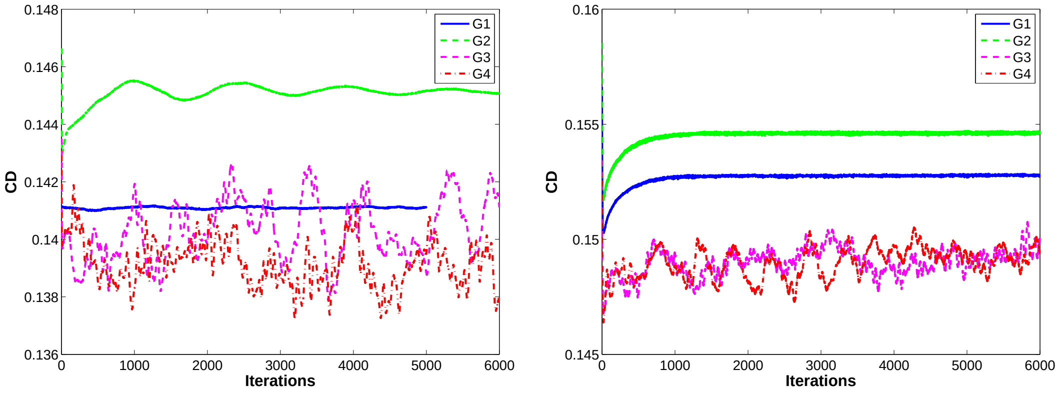
Figure 3.65: Third refined region for AoA = -10 *deg* in Z direction.

iterations after a converged solution has been reached and the following results are seen in Figure 3.66, Figure 3.67, and Figure 3.68 with results for both Mach numbers.



(a) Mach = 0.5 (b) Mach = 0.9

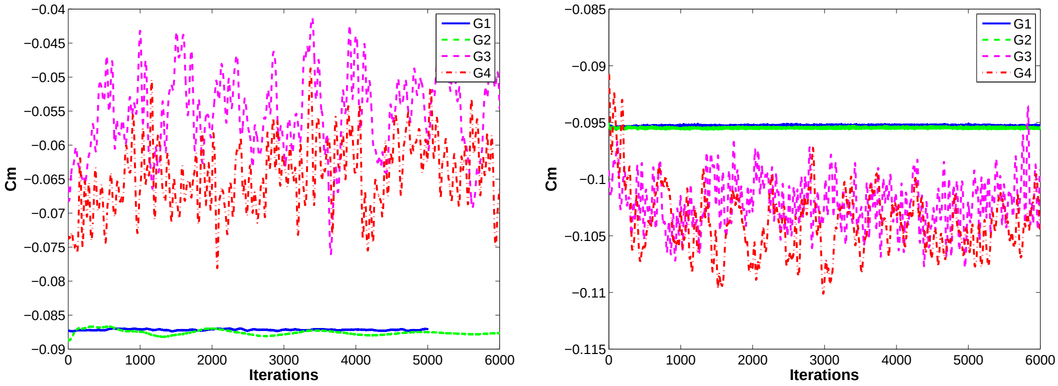
Figure 3.66: Grid refinement results for lift at -10 *deg* AoA.



(a) Mach = 0.5 (b) Mach = 0.9

Figure 3.67: Grid refinement results for drag at -10 *deg* AoA.

An interesting result happens in this grid convergence study; the change in the mean value from the first to the second grid is in the "opposite direction" from the jump from the first grid to the last two grids. It is seen that the oscillations of the coefficients in the first two grids is nearly zero when compared to the oscillations of



(a) Mach = 0.5 (b) Mach = 0.9

Figure 3.68: Grid refinement results for pitching moment −10 *deg* AoA.

the last two grids. This is due to the hypothesized result that the vorticity outboard on the wing appeared as though it "should" be unsteady but was not accurately being captured by the relatively coarse mesh in that region in the first two grids.

This "change in direction" of the mean coefficient values from Grid 1 to Grid 2 compared to Grid 1 to Grids 3 and 4 can happen when these fine oscillations in the flow begin to be captured which can affect the surface pressures and thereby affect the coefficients. These oscillations in the flow are important to model and include in the results obtained in an accurate CFD simulation. This constitutes much of the purpose for performing the grid refinement for this research when it was not purely necessary for the task at hand.

Grids 3 and 4 produced similar results where Grid 3 provides a cell count savings of almost 12 million. Due to the great computational savings and the similar results produced to the fine mesh, Grid 3 will be the final grid utilized for all maneuver calculations.

IV. Results

4.1 Results of Maneuvers

This section will outline the results of the CFD calculations performed for each Training Maneuver (TM) and Comparison Maneuver (CM). Not all TMs and CMs were finished in time for the presentation of these results. Table 4.1 shows which jobs were able to be accomplished at this time. The primary roadblock to finishing all cases was due to the computational expense. The large size of the jobs required the use of thousands of processors for up to days at a time as seen in Table 4.2. These large computational requirements equated into long wait times in the queuing system on Raptor for up to two weeks for one job to start. There were also software and hardware issues with Raptor which would at times delete jobs after they had been started, causing another long wait time in the queuing system. The combination of these two issues resulted in unsuccessful completion of many jobs. All cases but CM3 for the Mach 0.5 were accomplished, however, so focus is given to these results in this chapter.

Table 4.1: Summary of finished versus unfinished cases.

	Maneuver #							Comparison #		
Mach #	TM1	TM2	TM3	TM4	TM5	TM6	TM7	CM1	CM2	CM3
0.5	X	X	X	X	X	X	X	X	X	
0.9			X			X	X	X		X

Table 4.2: Summary of computational expense. Total number of hours ran = 1.64 million

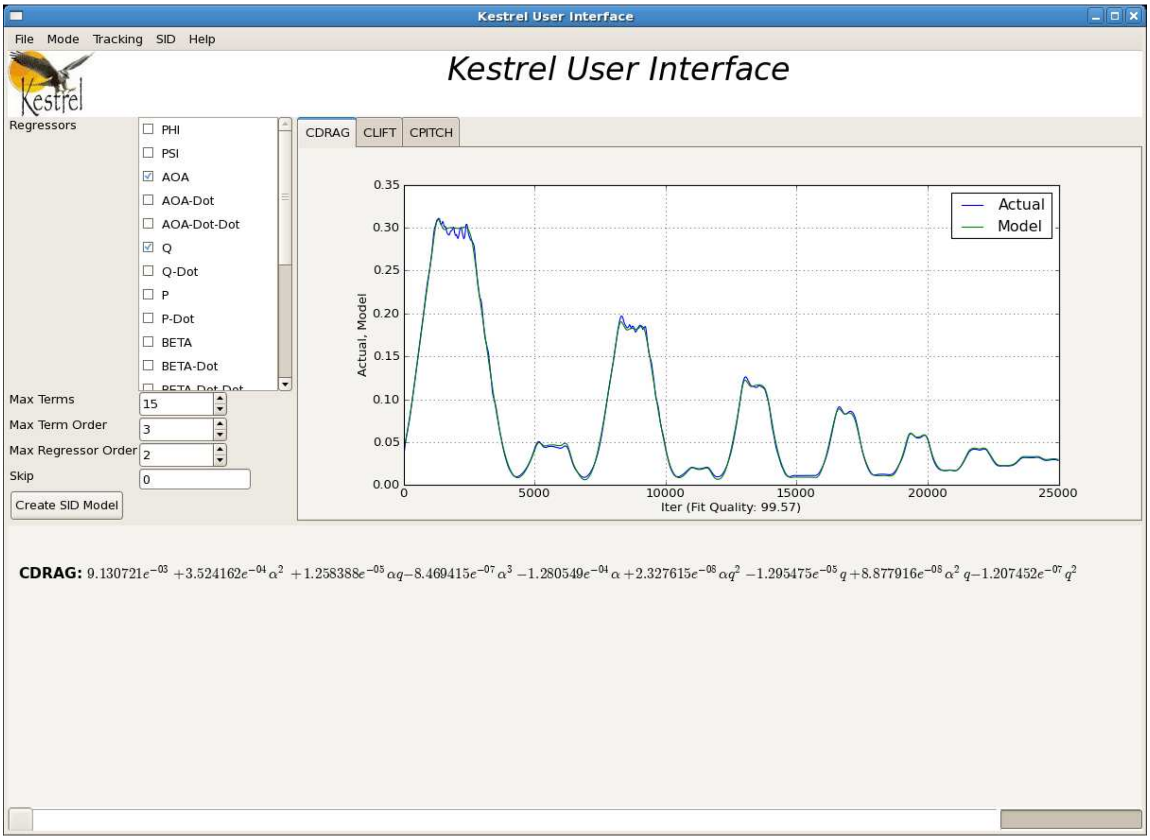
Mach #	CPU · Hours per Iter	# Iters	# Processors	# Hours per Job
0.5	2̃.5395	25000	2048	3̃1
0.9	2̃.0890	50000	2048	5̃1

Section 3.5 detailed the solver settings aside from the final value of the temporal damping terms. These terms should be lowered as far as possible for the most accurate solution. The static initial solutions of the flow field ended up with temporal advection damping coefficient of 0.2 for the Mach 0.9 case and 0.1 for the Mach 0.5 case. For

the TMs, the damping values were increased incrementally from those levels until a stable solution was reached. The final values for the Mach 0.9 and 0.5 cases were 0.4 and 0.3 respectively.

4.2 Created Models

Creating the reduced order models from each TM was done automatically in the Kestrel user interface as seen in Figure 4.1. The initial model created by the interface caps the "Max Term Order" and "Max Regressor Order" of the models at two for both. The maximum term order refers to the sum of all exponents for a given term and maximum regressor order refers to the highest exponent a regressor variable can have in a given term [12]. It was found that the models were quite insensitive to maximum regressor order, but quite sensitive to maximum term order. It is best



Regressors
☐ PHI
☐ PSI
☑ AOA
☐ AOA-Dot
☐ AOA-Dot-Dot
☑ Q
☐ Q-Dot
☐ P
☐ P-Dot
☐ BETA
☐ BETA-Dot
☐ BETA-Dot-Dot

Max Terms: 15
Max Term Order: 3
Max Regressor Order: 2
Skip: 0

CDRAG: $9.130721e^{-03} + 3.524162e^{-04}\alpha^2 + 1.258388e^{-05}\alpha q - 8.469415e^{-07}\alpha^3 - 1.280549e^{-04}\alpha + 2.327615e^{-08}\alpha q^2 - 1.295475e^{-05} q + 8.877916e^{-08}\alpha^2 q - 1.207452e^{-07} q^2$

Figure 4.1: Kestrel user interface for creation of models

to keep the model as simple as possible for improved ability to identify the model parameters which will result in the best predictive capability [15,21], but it must also include enough terms to accurately describe the response. The following equations represent the difference between two pitching moment coefficient predictions. The first is the predicted coefficient for a given TM with a maximum term order of two. The second is with a maximum term order of three.

$$C_m = C_0 + C_1\alpha + C_2Q + C_3\alpha^2 + C_4Q^2 + C_5\alpha Q \tag{4.1}$$

$$\begin{aligned} C_m = {} & C_0 + C_1\alpha + C_2Q + C_3\alpha^2 + C_4Q^2 + C_5\alpha Q \\ & + C_6Q + C_7\alpha^3 + C_8Q^3 + C_9\alpha^2Q + C_{10}\alpha Q^2 \end{aligned} \tag{4.2}$$

The models for lift and drag coefficients are exactly the same as the equations for pitching moment coefficient given in Eq. (4.1) and Eq. (4.2); only the coefficients vary. Higher orders than those can be utilized but the equations follow the same pattern with higher order terms added. It is not known ahead of time which maximum order of terms is “best” until they are validated with the comparison results. For this reason, models were created for each TM based on varying numbers of maximum term orders.

Here is a good place to re-emphasize the ability to simply take a derivative of the above equations with respect to a desired regressor variable to obtain a model for that coefficient. As an example, in Section 2.3 the importance of the slope of the pitching moment coefficient versus AoA curve is to longitudinal stability. To continue with this same example, a model for this derivative coefficient can be determined by taking the derivative of Eq. (4.1) or Eq. (4.2) with respect to AoA which produces the following:

$$C_{m,\alpha} = C_1 + 2 * C_3\alpha + C_5Q \tag{4.3}$$

$$C_{m,\alpha} = C_1 + 2 * C_3\alpha + C_5Q + 3 * C_7\alpha^2 + 2 * C_9\alpha Q + C_{10}Q^2 \tag{4.4}$$

where each of the coefficients are the same as those found in Eq. (4.1) or Eq. (4.2) respectively.

For the Mach 0.5 case, it was found that a gain was obtained from changing the maximum term order from two to three but minimal gain was obtained from allowing fourth order terms. Similarly for Mach 0.9, the change was significant between two and three maximum order terms but also between three and four. Therefore, the maximum term orders of two and three were both utilized in creating models for Mach 0.5 and two, three, and four for Mach 0.9.

4.3 Prediction of Comparison Results

4.3.1 Mach 0.5. The models created for all TMs and for each maximum term order option were all utilized in predicting the results of lift, drag, and pitching moment coefficients for each CM for their respective Mach numbers. Primary focus will be given to the Mach 0.5 cases as all TMs for this case were completed, even though only two of the three CMs were completed. Unfortunately, the CM left unfinished was the third one with the static data, which was the most important. Even still, important results can be obtained from the completed cases.

Figure 4.2 and Figure 4.3 show model predictions for lift coefficient of CM1. There is little variability between the seven models (created from the seven TMs) in predicting the results of the CMs. Figure 4.4 and Figure 4.5 show zoomed in sections of Figure 4.2; the first region being one of the peaks in the CM (close to 3.25 s) and the other being the small static region in this CM at -10 deg AoA (near 4 s). Each model performs quite well in predicting the peak of the lift curve shown in Figure 4.4 with the oscillations of CM1 due to the time accurate solutions of CFD where the models created filter these type of oscillations out of the TMs before they are made and would not capture these oscillations.

Again, very little variability is seen among the models in Figure 4.5 although the models created from TM3, TM4, TM5, and TM6 predict the static value calculated

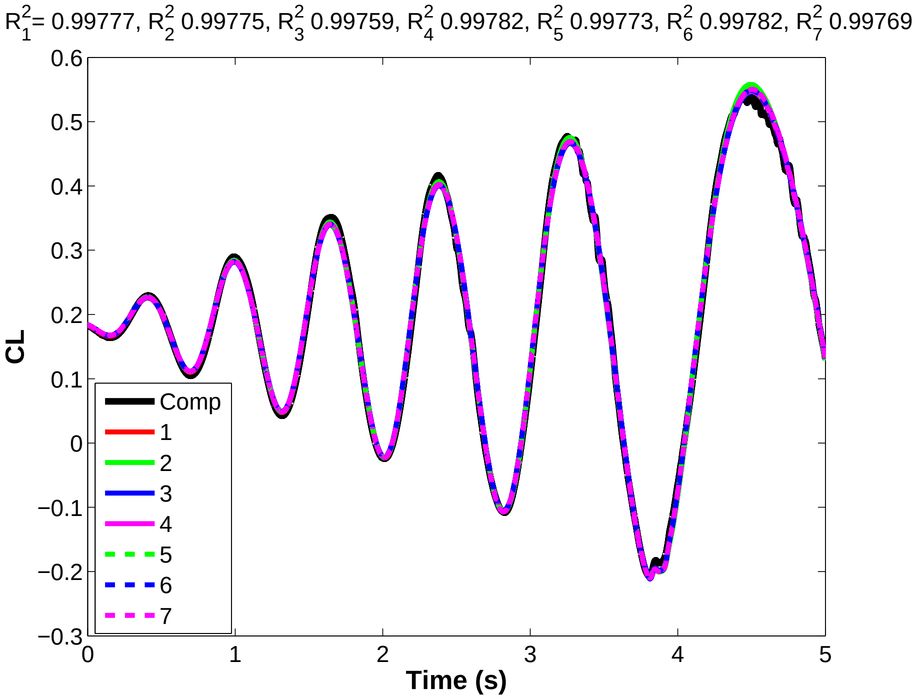


Figure 4.2: All models with maximum order of 2 predicting lift for CM1.

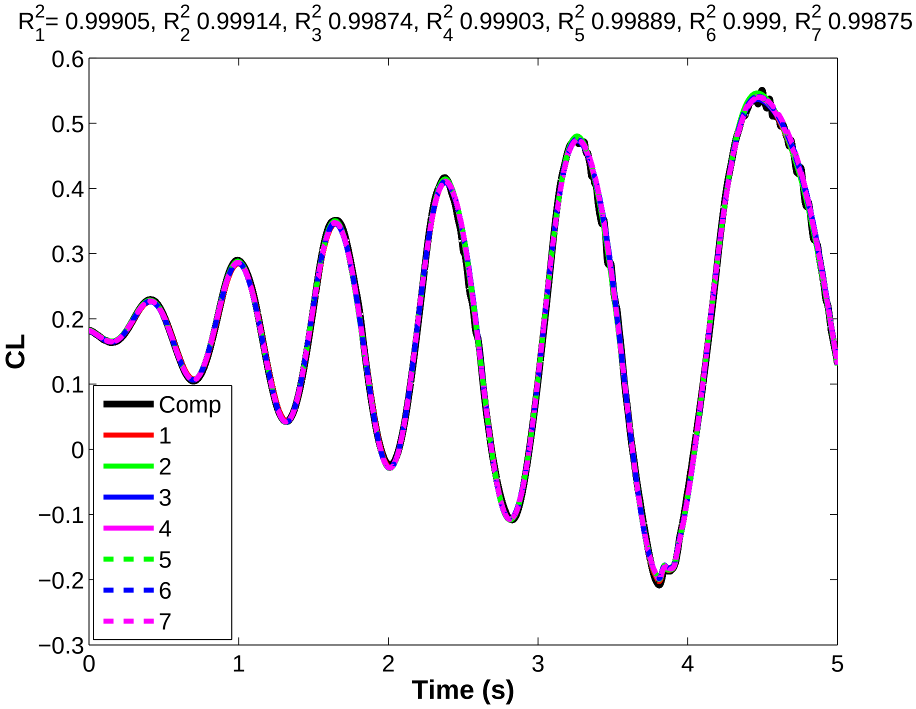


Figure 4.3: All models with maximum order of 3 predicting lift for CM1.

by the CM the best, but the difference is essentially negligible. This minor result is in accordance with the hypothesis that TMs with static regions – or at least low-pitch rate data in the case of TM6 – will produce better results in the static regions than the current best maneuver (TM2).

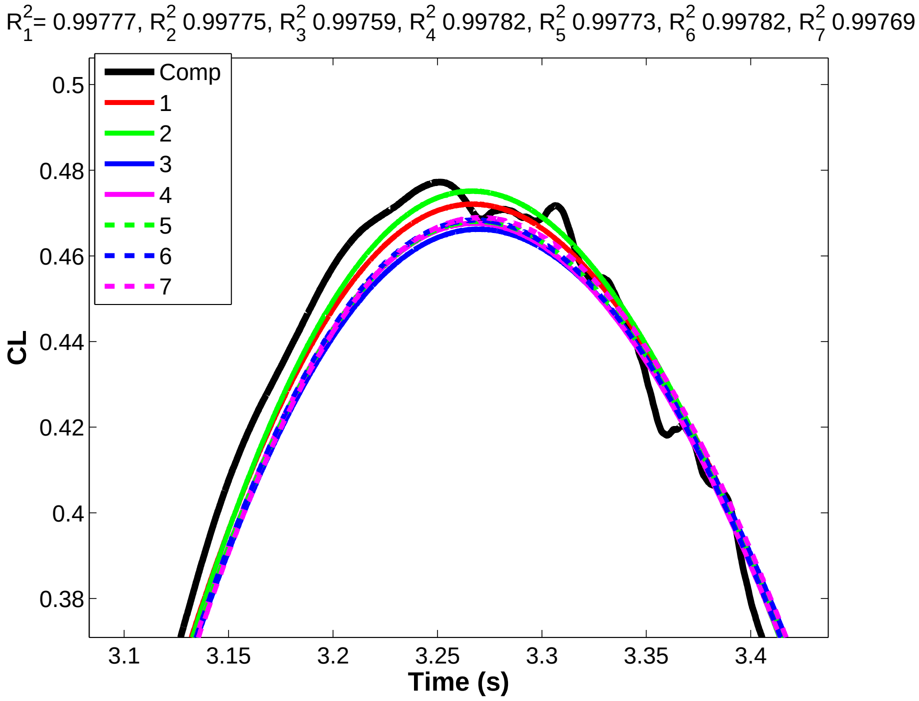


Figure 4.4: Zoomed in peak region of model predictions for CM1.

The drag coefficient predictions for CM1 are very similar to the lift coefficient. Figure 4.6 shows the predictions for the models with a maximum order of two for each term whereas Figure 4.7 is for a maximum order of three for each term. Very little variability among the models is seen in both of these figures, a trend which will continue for lift and drag coefficients.

Pitching moment coefficient, however, shows the most variability (as seen in Figures 4.8 - 4.9) in model prediction for all cases with TM1 and TM2 providing the worst predictions as indicated by the R^2 metric. All of the other TMs which have static and/or low pitch rate data at various AoA produce a better fit to the results. To help understand the flow field, a series of figures are provided of the CFD

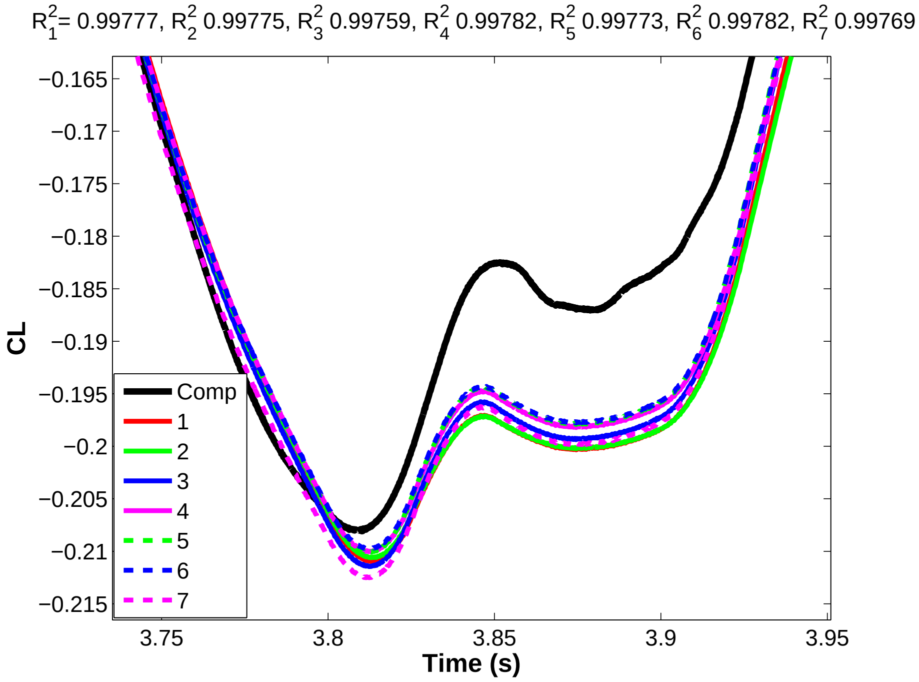


Figure 4.5: Zoomed in static region of model predictions for CM1.

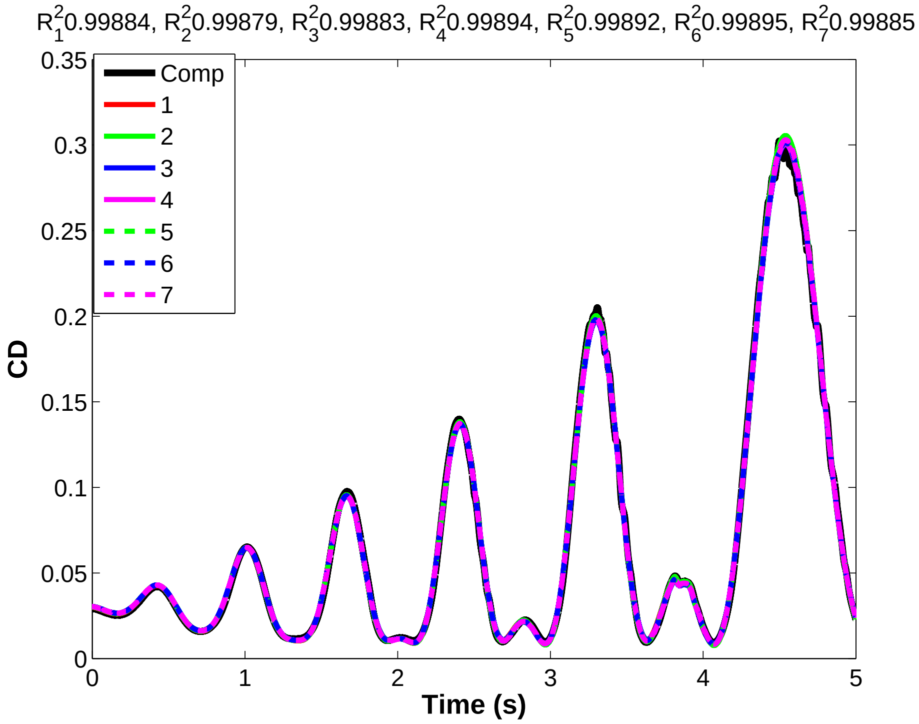


Figure 4.6: All models with maximum order of 2 predicting drag for CM1.

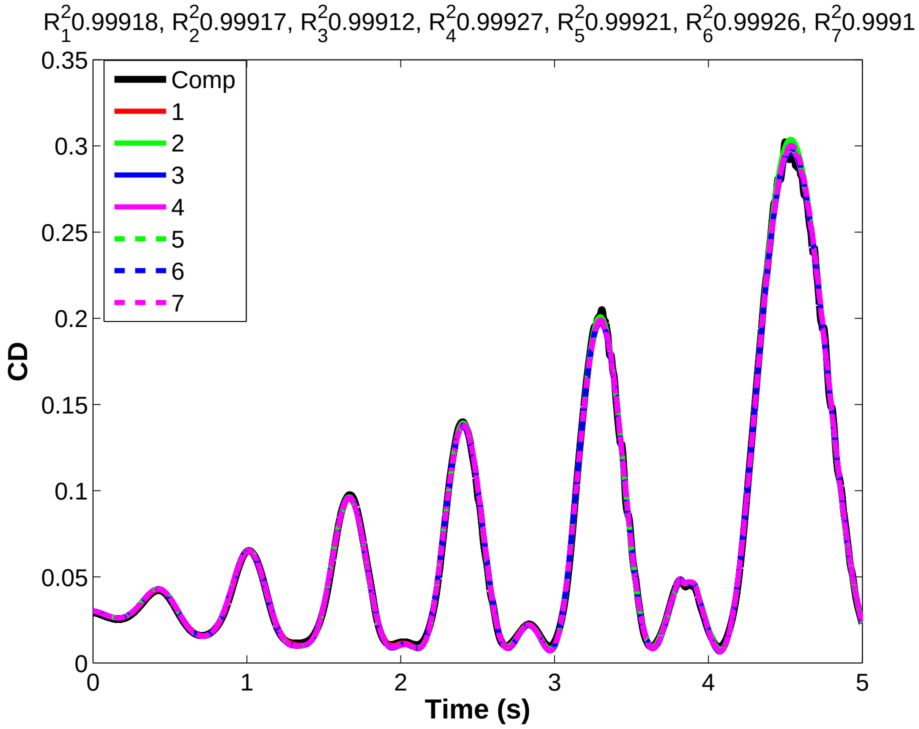


Figure 4.7: All models with maximum order of 3 predicting drag for CM1.

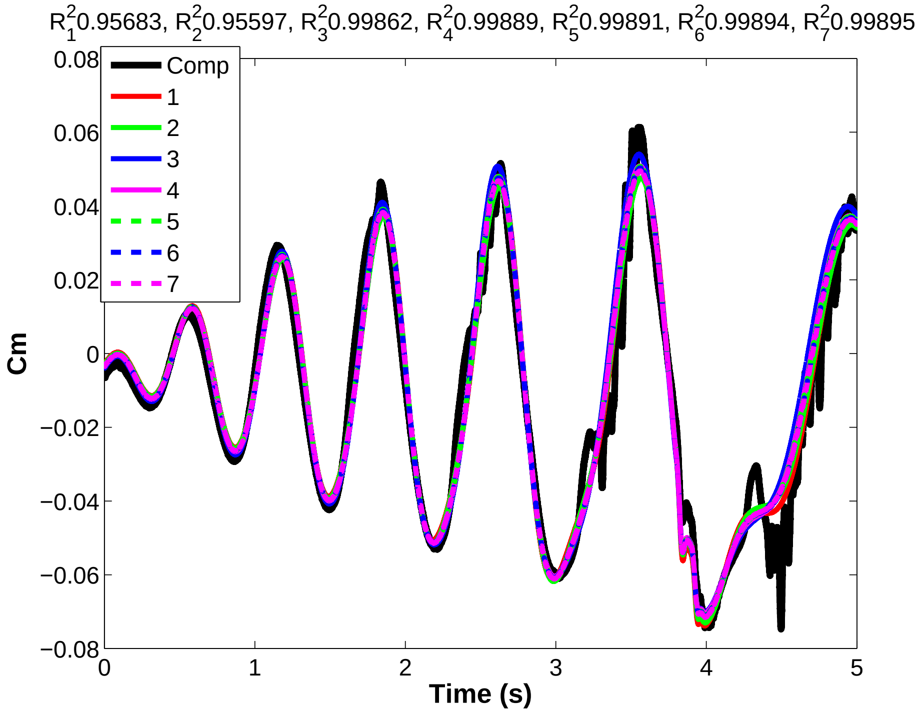


Figure 4.8: All models with maximum order of 2 predicting pitching moment for CM1.

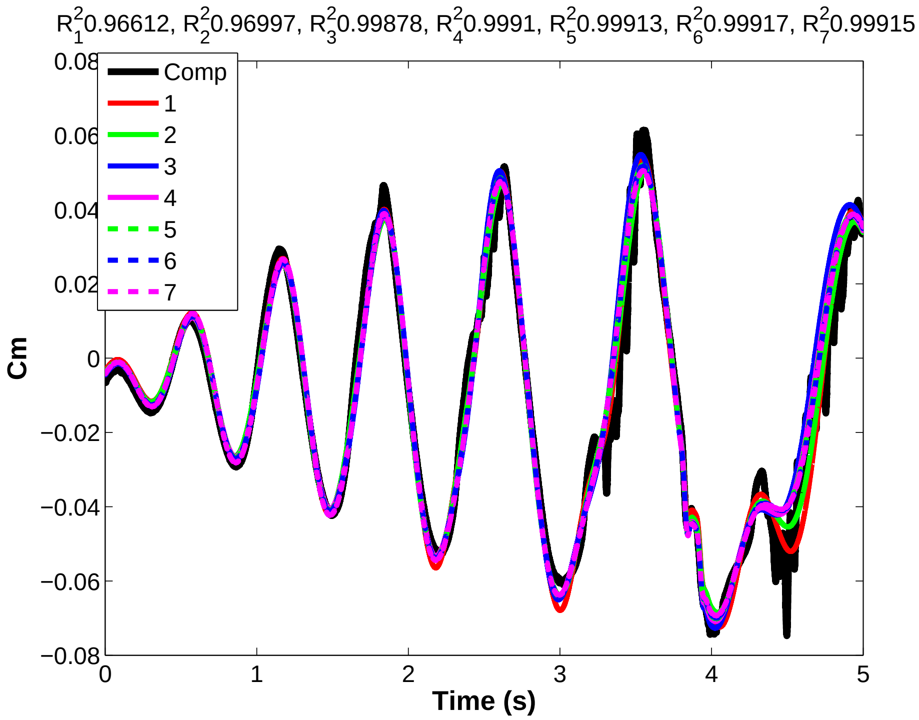


Figure 4.9: All models with maximum order of 3 predicting pitching moment for CM1.

solution to the flow over the aircraft as the maneuvers are performed. The first figure, (Figure 4.10), shows the Q-Threshold iso-surface (which relates to unsteadiness and separation) of CM1 at the end of the short static region found at -10 *deg* AoA (about at $t = 4s$) as compared to the flow solution of the static simulation (Figure 4.11) on the same grid, again at -10 *deg* AoA. There are only subtle differences between these two plots as they are both at least close to static data (the CM1 figure has an instantaneous pitch rat of 1.77 $\frac{deg}{s}$) comparisons even though one is during the course of a prescribed motion. In Figure 4.12, however, the flow is in a portion of highest pitch rate (98.7 $\frac{deg}{s}$) at 8 *deg* AoA with the same iso-surface as plotted in Figures 4.10 - 4.11. This is similarly compared to the static case for the same grid, again at 8 *deg* AoA. The iso-surfaces on these two plots are vastly different. The differences being derived from the high, positive pitch rate and its corresponding affect on the flow field and hence the coefficients.

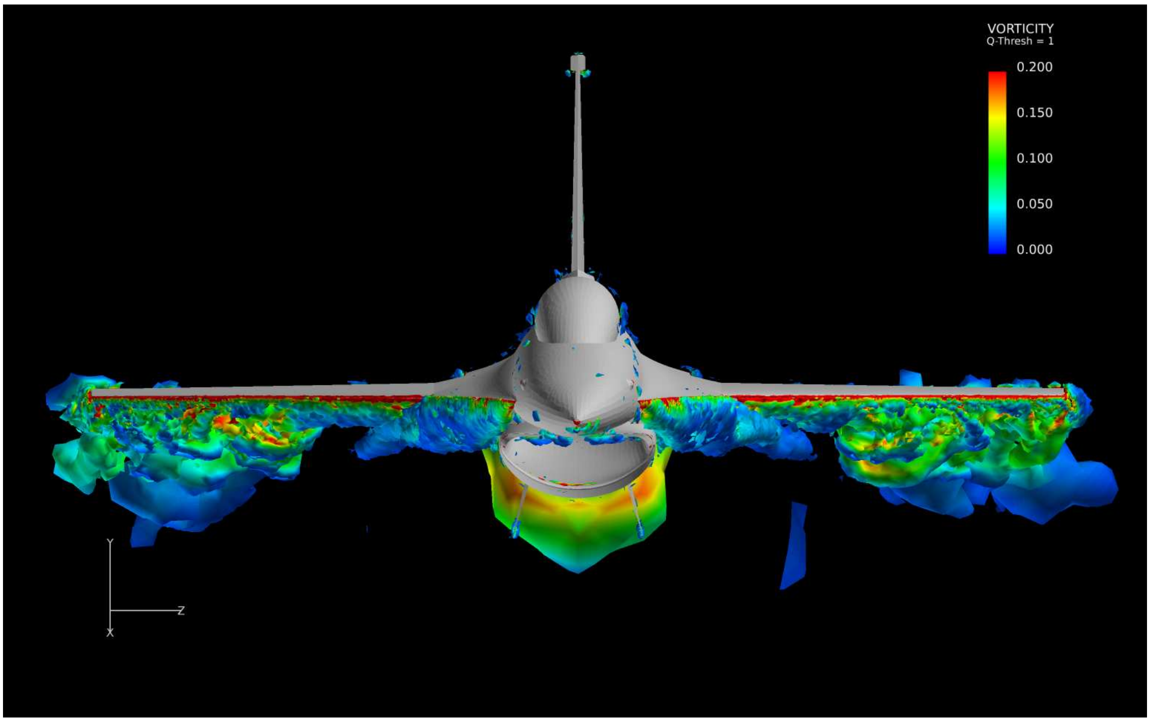


Figure 4.10: Flow field of CM1 near zero pitch rate at -10 *deg* AoA.

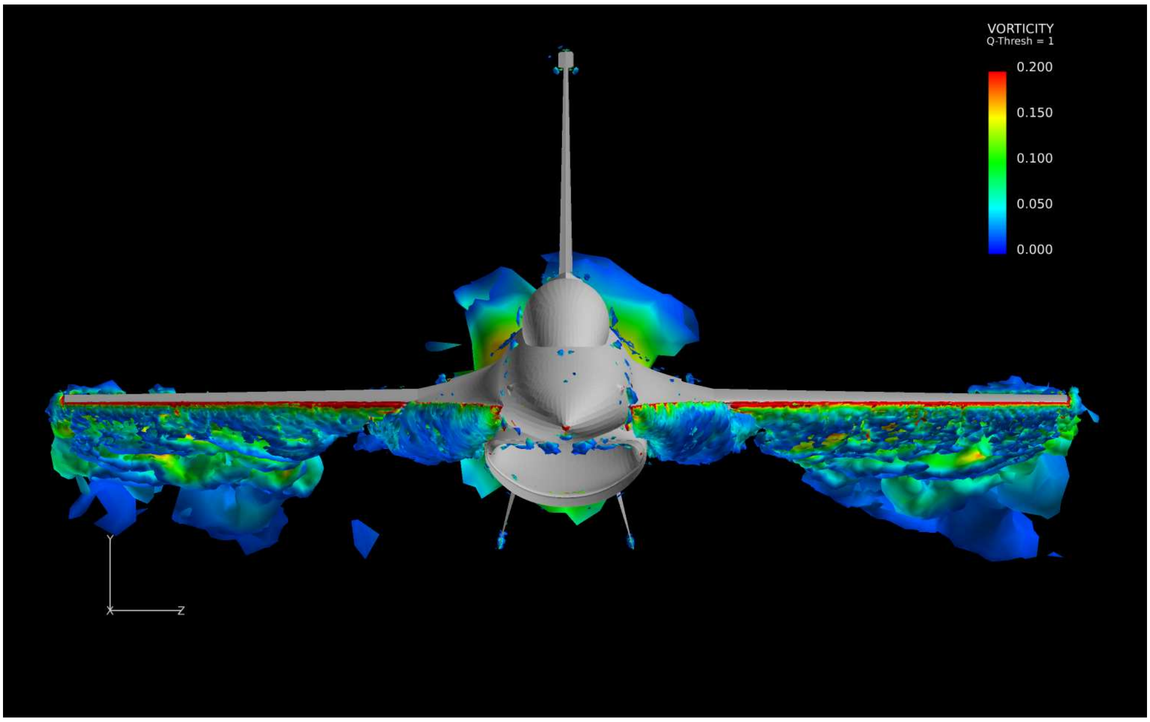


Figure 4.11: Flow field of static solution at $m10$ *deg* AoA.

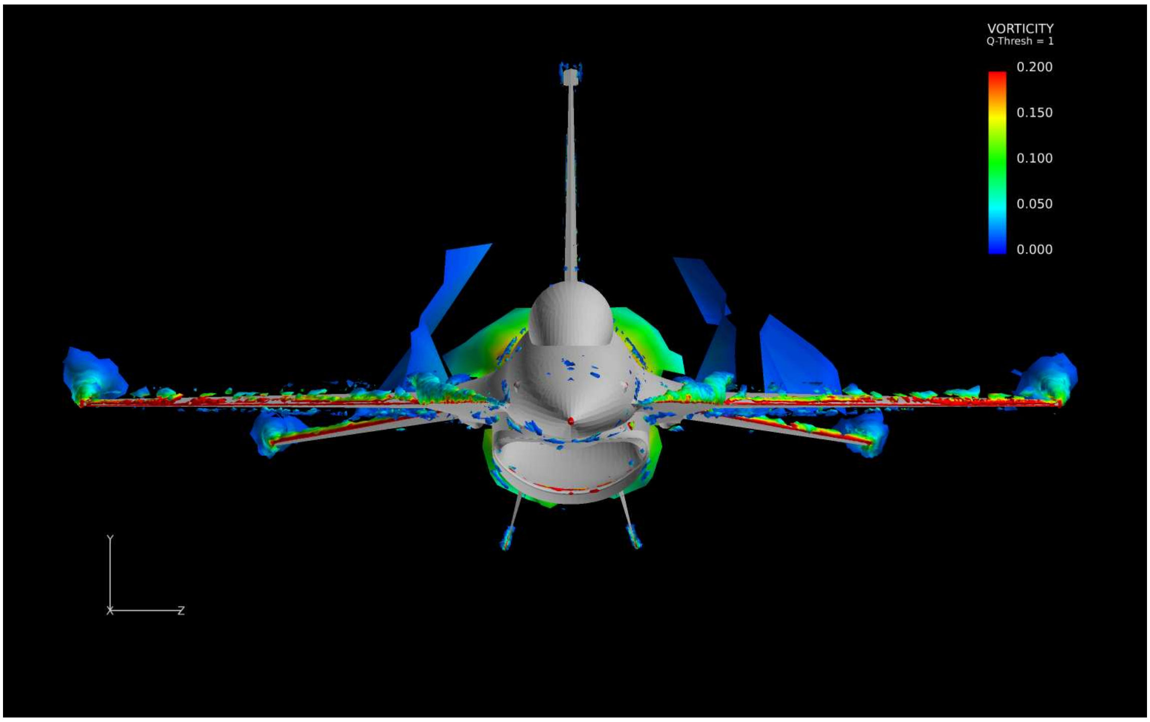


Figure 4.12: Flow field of CM1 at high, positive pitch rate 8 *deg* AoA.

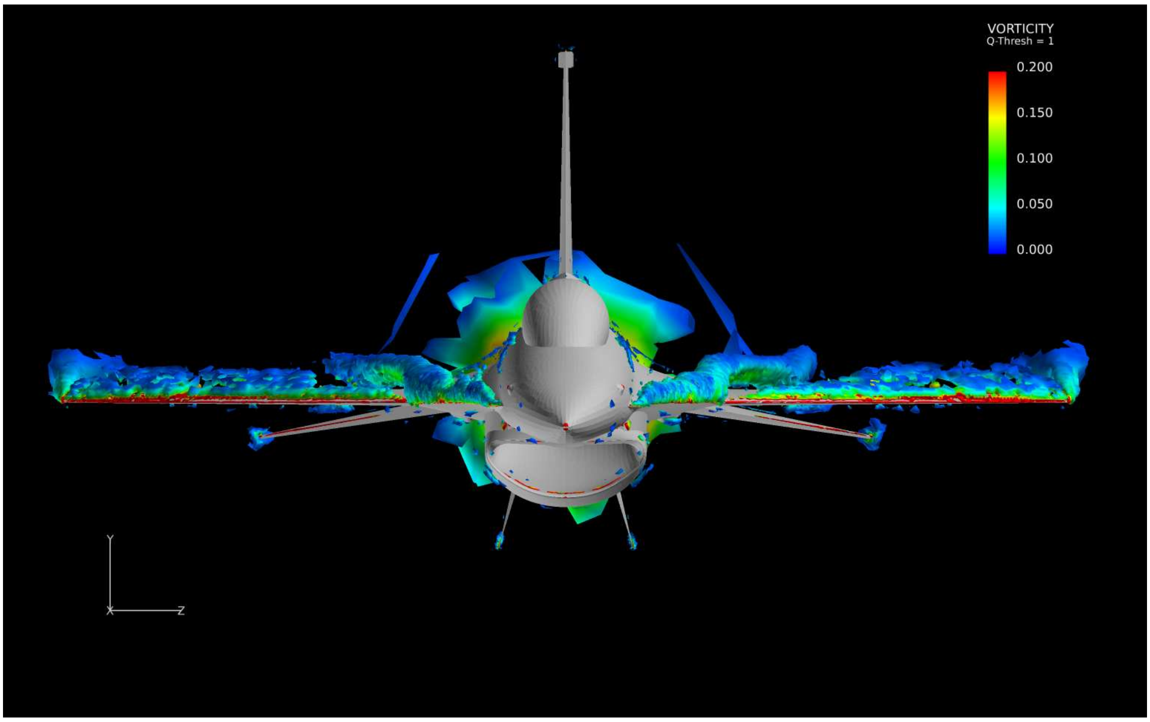


Figure 4.13: Flow field of static solution at 8 *deg* AoA.

Similar to the model predictions for CM1, the lift and drag predictions are all comparable for CM2 as seen in Figure 4.14 through Figure 4.17. Also similar to CM1 are the results of CM2. Seen in Figure 4.18 and Figure 4.19 are the pitching moment predictions where Models 3 – 7 compare better than 1 and 2.

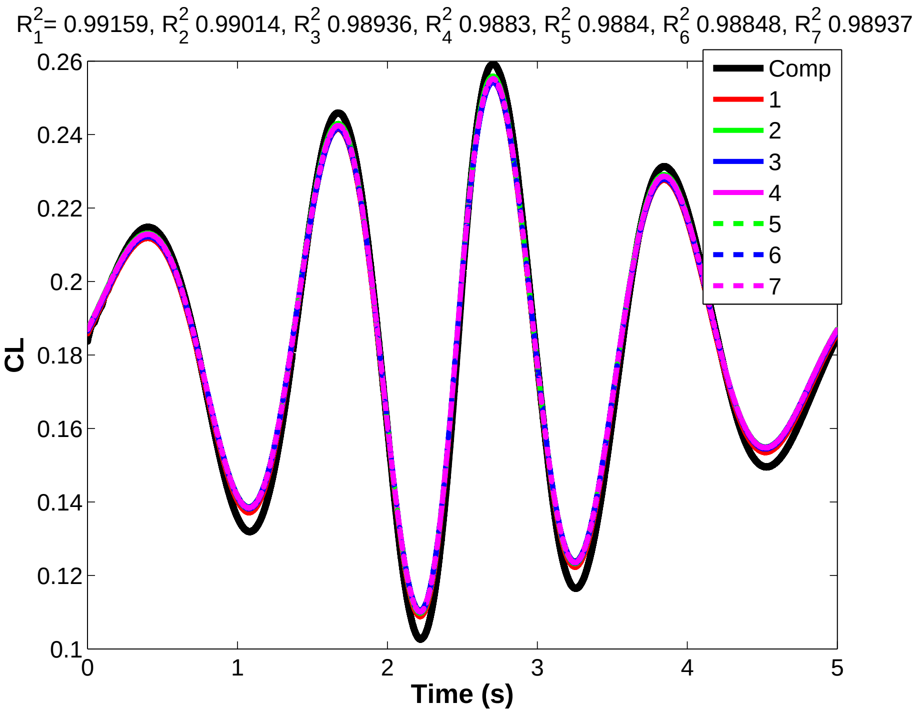


Figure 4.14: All models with maximum order of 2 predicting lift for CM2.

Due to the similar results for lift and drag prediction from the various models, plots are given of the static lift, drag, and pitching moment coefficients as predicted by each model. These plots are seen in Figures 4.20 - 4.22, and the last plot of pitching moment also includes a model prediction of pitching moment slope coefficient ($C_{m,\alpha}$) for only TM1 as an example of how simple, yet accurate, this process is (as discussed in Section 4.2 and whose models are shown in Eq. (4.3) and Eq. (4.4)).

Figures 4.20 - 4.22 assure the results seen in the comparison data plots, and provide a means to pull conclusions about the static “goodness” of each model without the results of CM3. This is only possible since the variability in lift and drag versus AoA among all models is very little. Because of this, it can be stated that for the

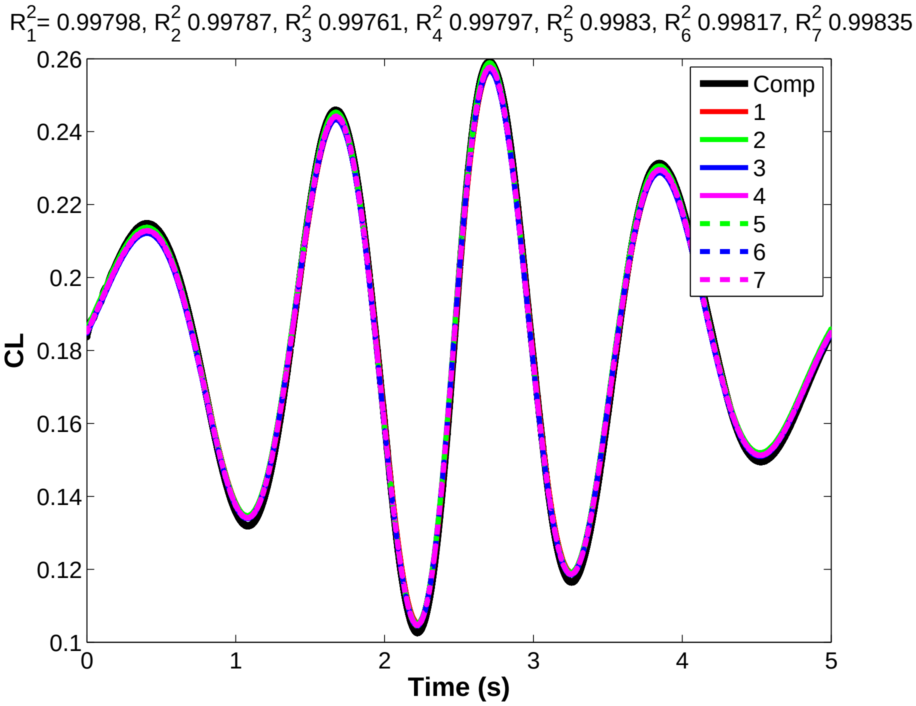


Figure 4.15: All models with maximum order of 3 predicting lift for CM2.

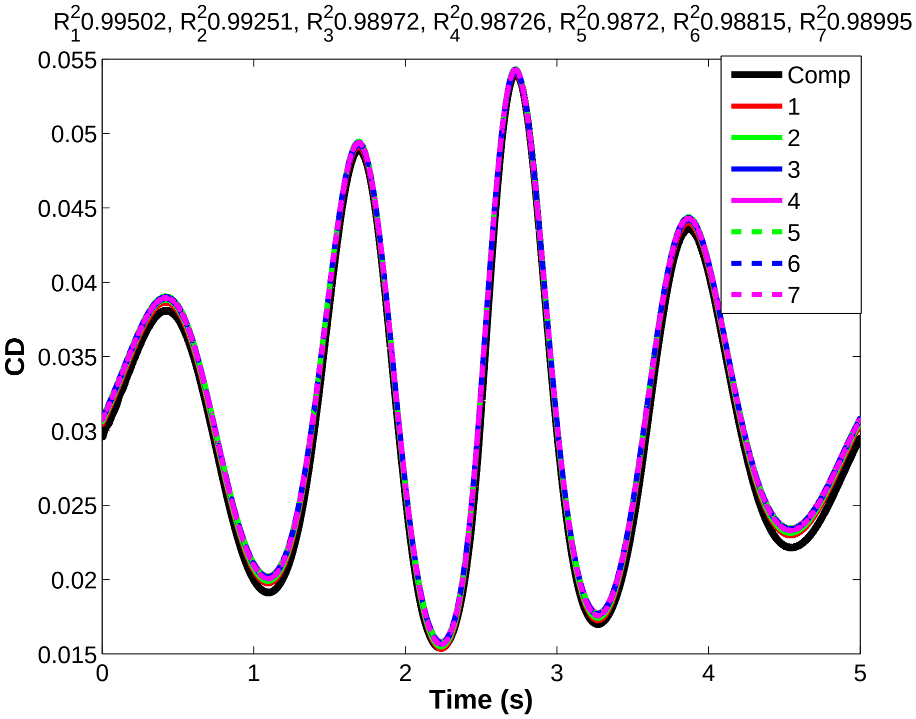


Figure 4.16: All models with maximum order of 2 predicting drag for CM2.

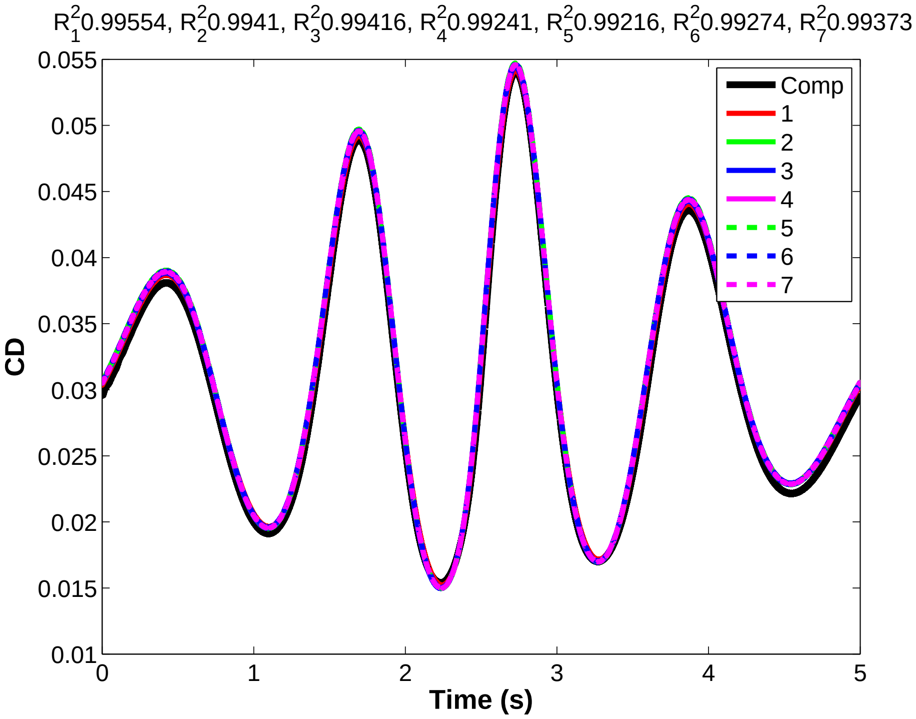


Figure 4.17: All models with maximum order of 3 predicting drag for CM2.

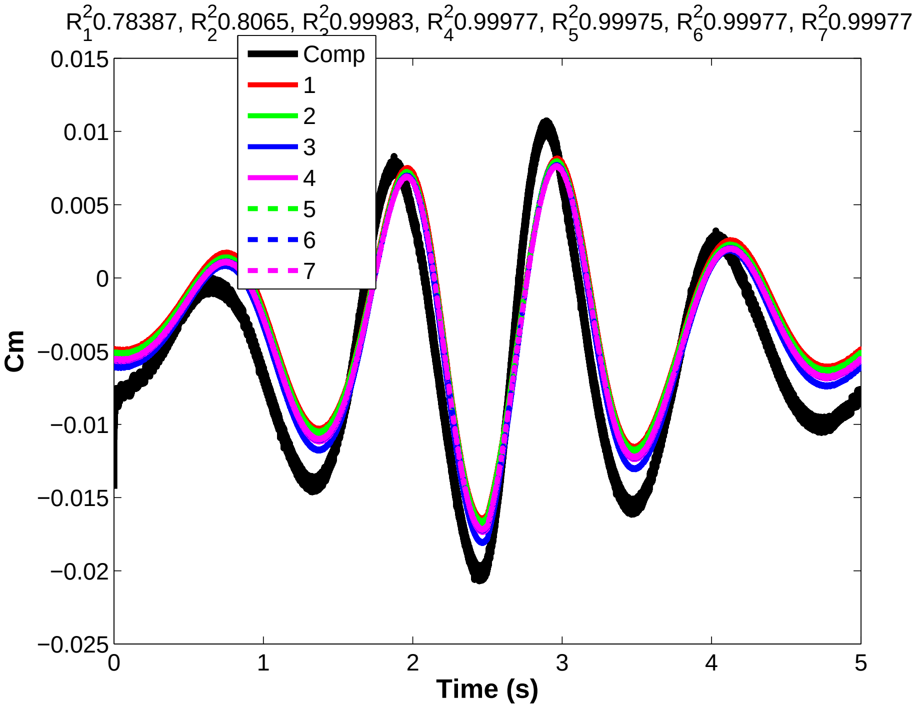


Figure 4.18: All models with maximum order of 2 predicting pitching moment for CM2.

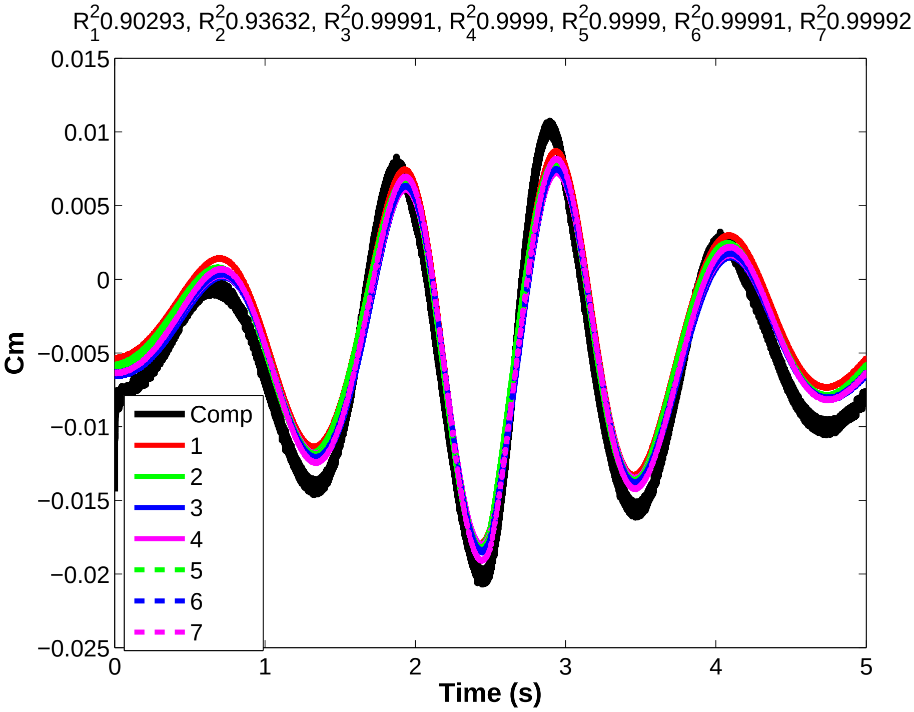


Figure 4.19: All models with maximum order of 3 predicting pitching moment for CM2.

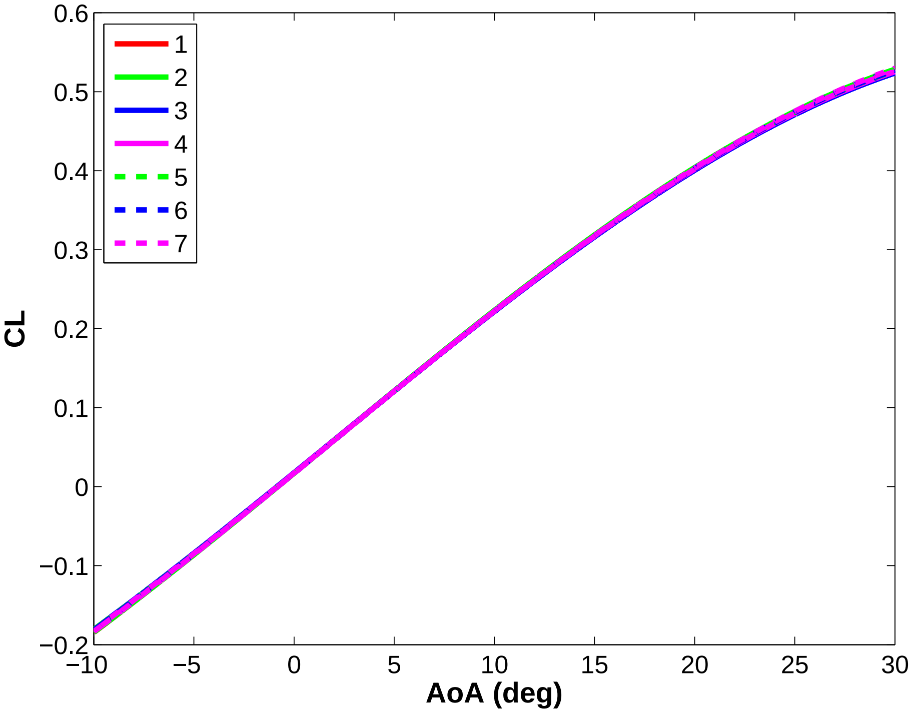


Figure 4.20: Comparison of all models predicting static C_L versus AoA curve.

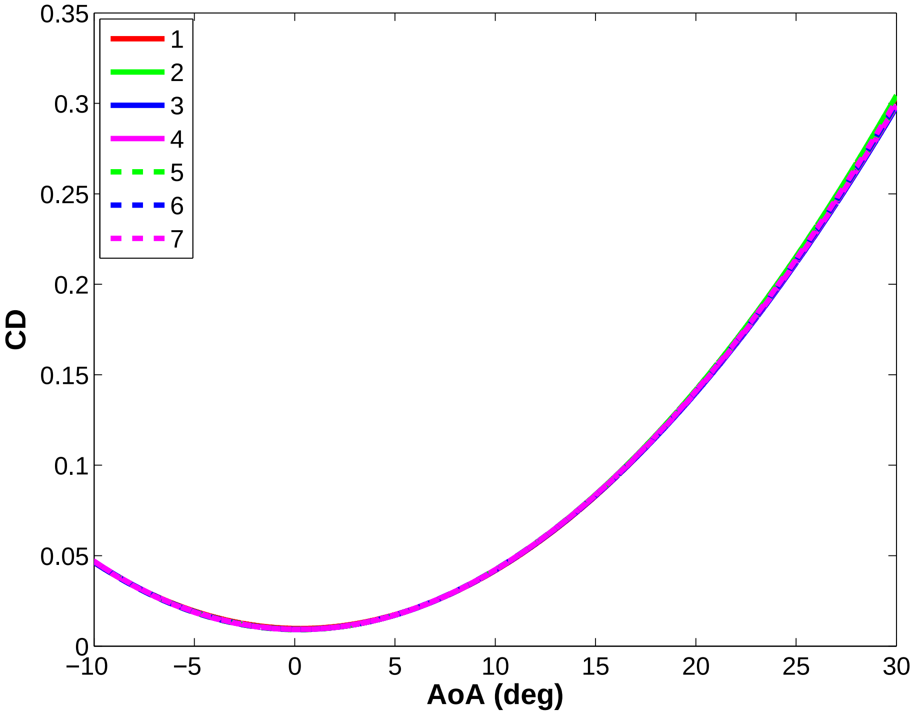


Figure 4.21: Comparison of all models predicting static C_D versus AoA curve.

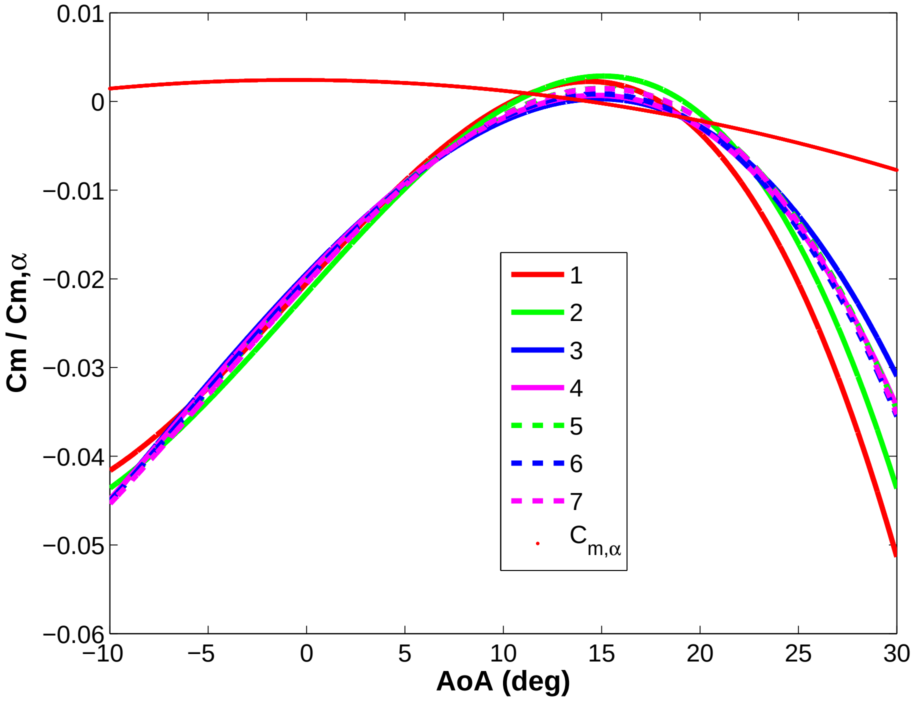


Figure 4.22: Comparison of all models predicting static C_m versus AoA curve.

given regressor space, flight condition, optimization constraints, and TMs, the benefit of the addition of static and low pitch rate data is of little importance or no harm in the prediction of static lift or drag coefficients. If there was large variability, CM3 would be needed to determine which one fits the results the best.

The difference in the pitching moment versus AoA, however, is drastic when comparing TM1 and TM2 to the rest of the TMs (see Figure 4.22). This may in large part be due to the unsteadiness in pitch moment coefficient that is introduced at high pitch rate as well as the non-linear AoA values. This can be seen in CM1 as amplitude is ramped (with a corresponding pitch rate increase), oscillations in pitch rate in the linear region appear. The start of oscillations are seen in Figure 4.18 and Figure 4.19 along the $C_m = 0$ line comparing before versus after $t = 2s$. The $C_m = 0$ line corresponds to an AoA of about 8 deg - definitely in the linear region. Therefore, having static regions in the flow allows for smoother data points at the various AoA values, producing better predictability of the mean value. From this, it can be determined that the addition of static and low pitch rate data is of significant benefit to the prediction of pitching moment coefficient.

It was also determined that the predictions of Models 3, 4, 5, and 6 produced the best results for pitching moment and Figure 4.22 is a good comparison for how much worse the predictions of the other models are than the ones just listed. From Tables 3.2 - 3.6, the metrics which these TMs excel are generally metrics 5 - 10: the metrics describing zero and low pitch rate characteristics.

From Figures 4.2 - 4.19, both maximum term order choices were plotted. A comparison between the better of the two is seen in those figures that the maximum term order of three consistently produced better results. This can also be seen by comparing Table 4.3 and Table 4.4. Table 4.3 shows the R^2 values for the models created with an allowed maximum term order of two. In every instance, these values are smaller (worse) than those found in Table 4.4 which are models created from the

Table 4.3: R^2 metrics for each model with maximum term order of two.

Maneuver	C_L		C_D		C_m	
	CM1	CM2	CM1	CM2	CM1	CM2
TM1	0.99777	0.99159	0.99884	0.99502	0.95683	0.78387
TM2	0.99775	0.99014	0.99879	0.99251	0.95597	0.80650
TM3	0.99759	0.98936	0.99883	0.98972	0.99862	0.99983
TM4	0.99782	0.98830	0.99894	0.98726	0.99889	0.99977
TM5	0.99773	0.98840	0.99892	0.98720	0.99891	0.99975
TM6	0.99782	0.98848	0.99895	0.98815	0.99894	0.99977
TM7	0.99769	0.98937	0.99885	0.98995	0.99895	0.99977

Table 4.4: R^2 metrics for each model with maximum term order of three.

Maneuver	C_L		C_D		C_m	
	CM1	CM2	CM1	CM2	CM1	CM2
TM1	0.99905	0.99798	0.99918	0.99554	0.96612	0.90293
TM2	0.99914	0.99787	0.99917	0.99410	0.96997	0.93632
TM3	0.99874	0.99761	0.99912	0.99416	0.99878	0.99991
TM4	0.99903	0.99797	0.99927	0.99241	0.99910	0.99990
TM5	0.99889	0.99830	0.99921	0.99216	0.99913	0.99990
TM6	0.99900	0.99817	0.99926	0.99274	0.99917	0.99991
TM7	0.99875	0.99835	0.99910	0.99373	0.99915	0.99992

same TM with the only difference being the allowed maximum term order was allowed to be three.

The results of the Mach 0.5 case display a lack of variability between the models created from the various TMs in lift and drag coefficients. This may be due to the linear nature of much of the regressor space defined. Perhaps the AoA range should have been extended beyond 30 *deg* AoA. It could also be that the time it takes for the transients in the flow to die out after zero pitch rate is held, has significant effects on the modeled values. While the transients are "dying out," the pitch rate is still at zero so the model coefficients are trying to capture the transient magnitudes of the coefficients as well as the static values and the result is seen as little difference from the predictions of the completely dynamic cases. The last theory for the lack of variability is the possibility that the length of time allowed for the TMs in this research was too long, resulting in a situation where even the "worst" TMs were able to cover the regressor space enough to still provide an accurate model of the region.

Ultimately, no specific TM stands out as the best and without clear distinction between maneuvers, no clear conclusion can be made about metric validation. As stated above, the pitching moment prediction shows a correlation to the zero and low pitch rate metrics so the hypothesis of this region of data being of great importance is upheld, but more testing is required before validation/invalidation can be deemed. It is unclear which region of low pitch rate is most important or if static data is more important than this low pitch rate data or maybe they are equally important? Such conclusions cannot be made as of yet, just that initial results support the original hypothesis.

It is determined, however, that the included effects of the higher-order terms of the regressor variables when creating the models produces better predictability for both CMs examined herein. Also, the ability of the models with static or extra low pitch rate data were able to predict the dynamic regions of the comparison data sets nearly as well as the current best TM. This strong conclusion helps to satisfy the concern about the trade-off of adding this static data in place of extra dynamic coverage of the regressor space.

4.3.2 Mach 0.9. Only half of the total number of TMs and CMs were able to finish in time for this thesis. Unfortunately, TM2 (the current best) was not among those completed so it is difficult to draw conclusions beyond those already obtained from the first Mach number, especially since the entire flight envelope at this Mach number of 0.9 is nonlinear as it is in transonic flow.

Despite those short-comings, the data gathered is still of much worth. CM3 with the static data was completed for this Mach number, while it was not for Mach 0.5. The comparisons of the three models created from TMs 3, 6, and 7 to CM1 are very similar to the results seen for Mach 0.5 so these plots will not be shown. The plots for lift, drag, and pitching moment coefficient are found in Figure 4.23, Figure 4.24, and Figure 4.25 respectively. Only the models built using maximum term order of three are shown as they all produced similar results but these cases produce the best R^2 metrics on average as seen in Table 4.5 through Table 4.7. The pitching moment prediction is much worse over all than the predictions of the other CM or for pitch moment predictions for the other Mach number. The same models predict the

Table 4.5: R^2 metrics for each model with maximum term order of two.

	C_L		C_D		C_m	
Maneuver	CM1	CM3	CM1	CM3	CM1	CM3
TM3	0.99538	0.94839	0.99781	0.99150	0.99927	0.94938
TM6	0.99544	0.94920	0.99788	0.99131	0.99930	0.94929
TM7	0.99545	0.94934	0.99790	0.99112	0.99931	0.94809

Table 4.6: R^2 metrics for each model with maximum term order of three.

	C_L		C_D		C_m	
Maneuver	CM1	CM3	CM1	CM3	CM1	CM3
TM3	0.99751	0.96972	0.99951	0.99708	0.99929	0.95766
TM6	0.99752	0.96946	0.99954	0.99670	0.99931	0.95724
TM7	0.99744	0.96951	0.99954	0.99669	0.99932	0.95664

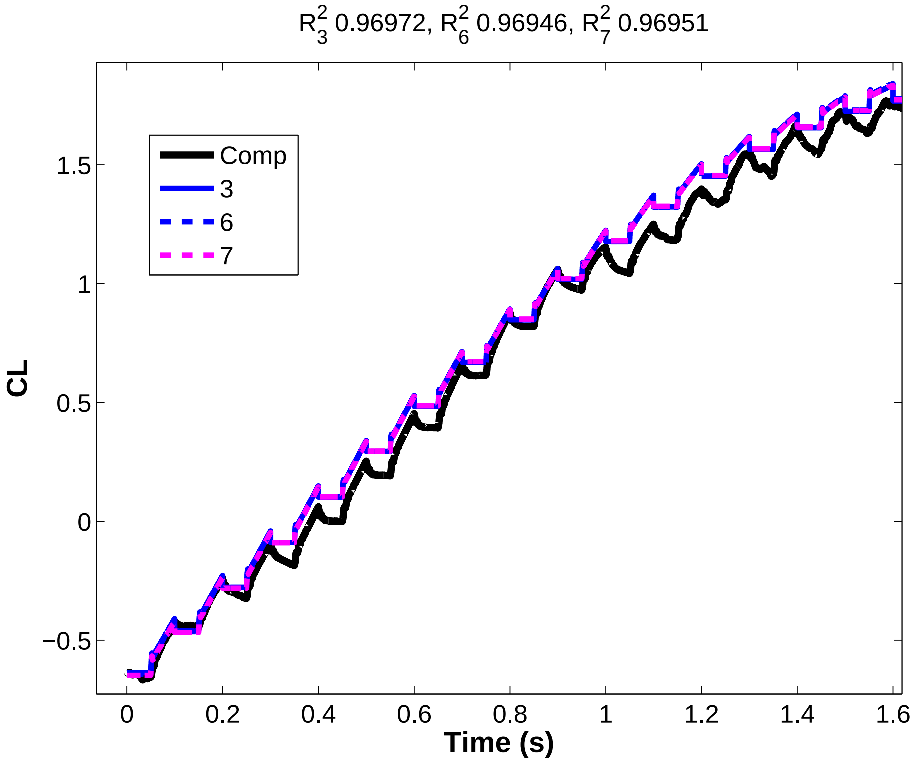


Figure 4.23: All models with maximum order of 3 predicting lift for CM3.

Table 4.7: R^2 metrics for each model with maximum term order of four.

Maneuver	C_L		C_D		C_m	
	CM1	CM3	CM1	CM3	CM1	CM3
TM3	0.99854	0.98225	0.99972	0.99542	0.99968	0.93117
TM6	0.99860	0.98260	0.99971	0.99558	0.99968	0.93286
TM7	0.99843	0.98175	0.99970	0.99579	0.99966	0.93528

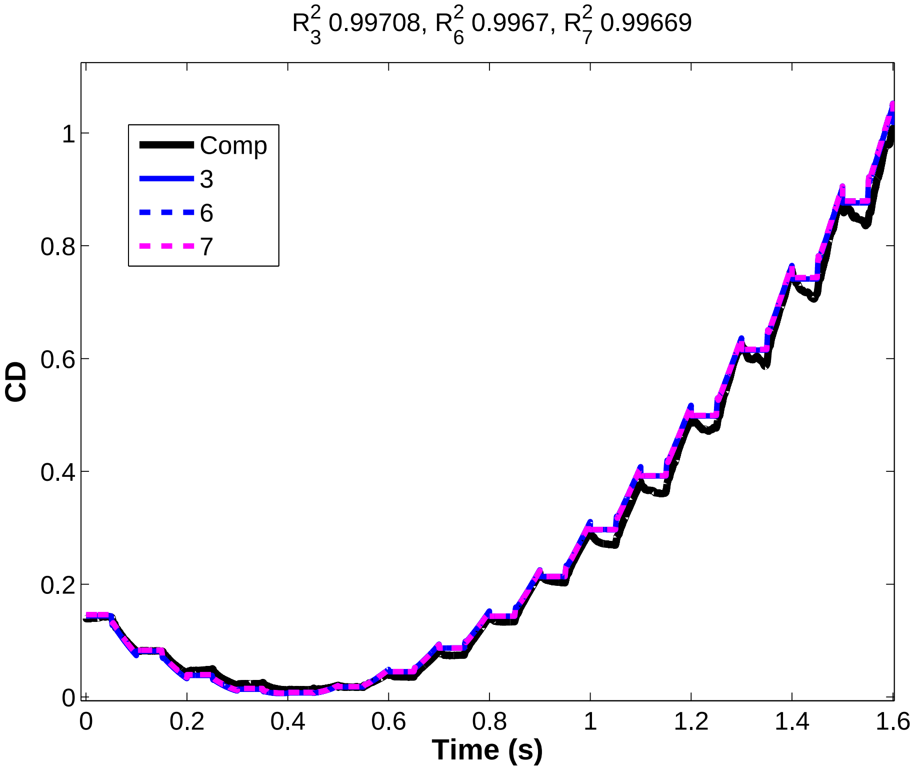


Figure 4.24: All models with maximum order of 3 predicting drag for CM3.

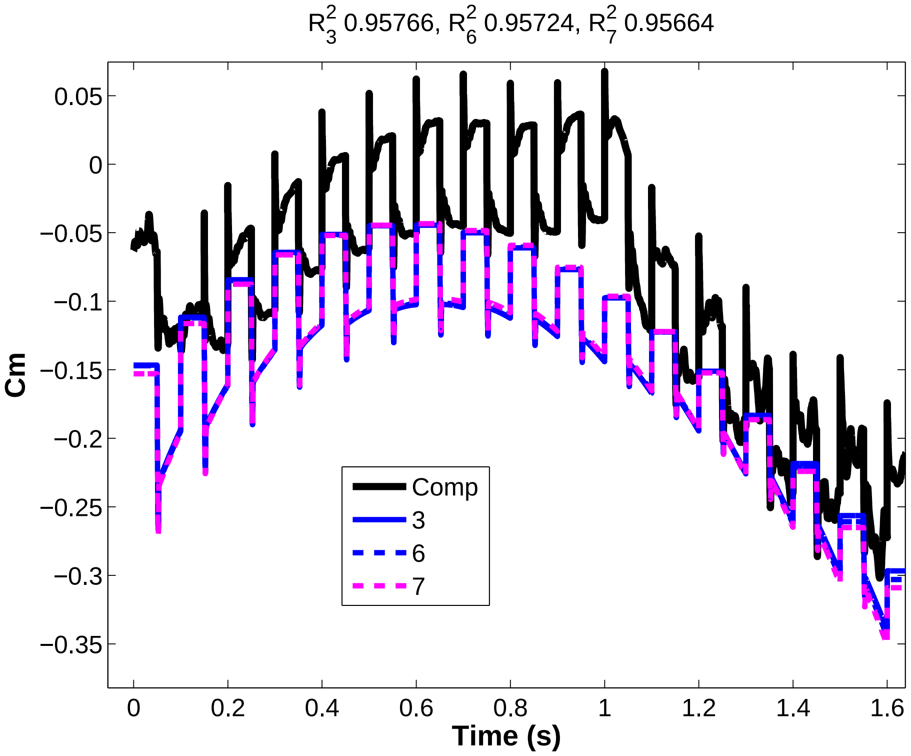


Figure 4.25: All models with maximum order of 3 predicting pitching moment for CM3.

pitching moment coefficient for CM1 just as well as the models at Mach 0.5, so one theory is this may be due to the large accelerations of CM3 from static to dynamic and then back again to static. This would have to be tested through the creation of a different "static" maneuver - one which will smooth the corners and hence decrease the accelerations seen.

Plots for the static coefficient predictions of the three TMs are also given for comparison to Mach 0.5 data. These plots are seen in Figure 4.26, Figure 4.27, and Figure 4.28. These plots are given with the maximum regressor term order of three just as with the other plots at this Mach 0.9 case. The lift and drag coefficients are again, very similar to one another although the first two TMs are not included in these results so it is possible variation among those may occur. The pitching moment coefficient for even these three TMs is seen to the same extent as is seen in Figure 4.22 among these TMs so it may be theorized that the additions of the static regions in the flow field would again help to produce better modeling of the pitching moment

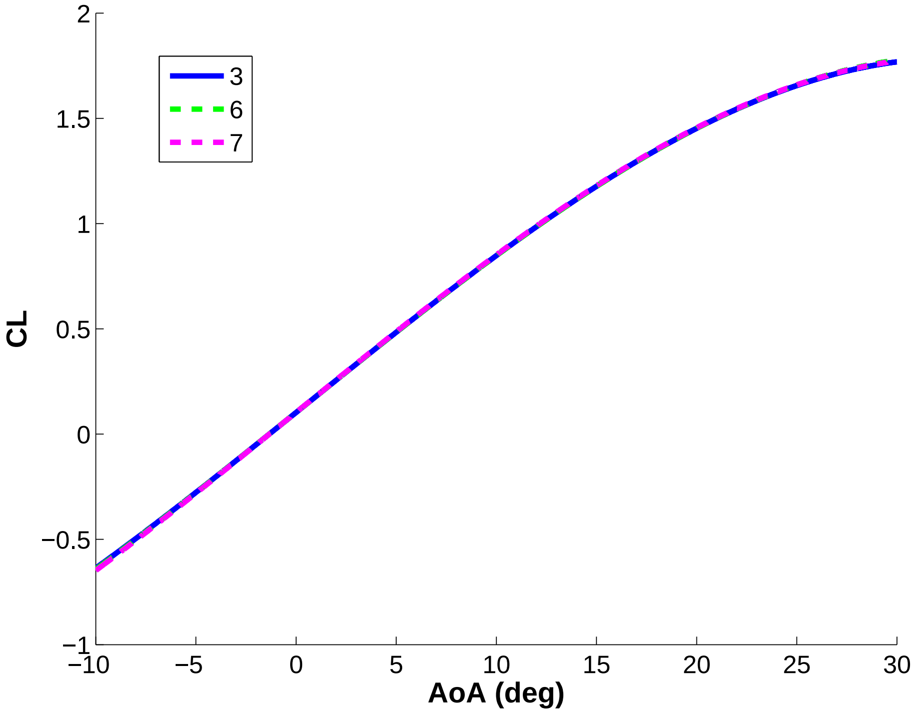


Figure 4.26: Comparison of all models predicting static C_L versus AoA curve.

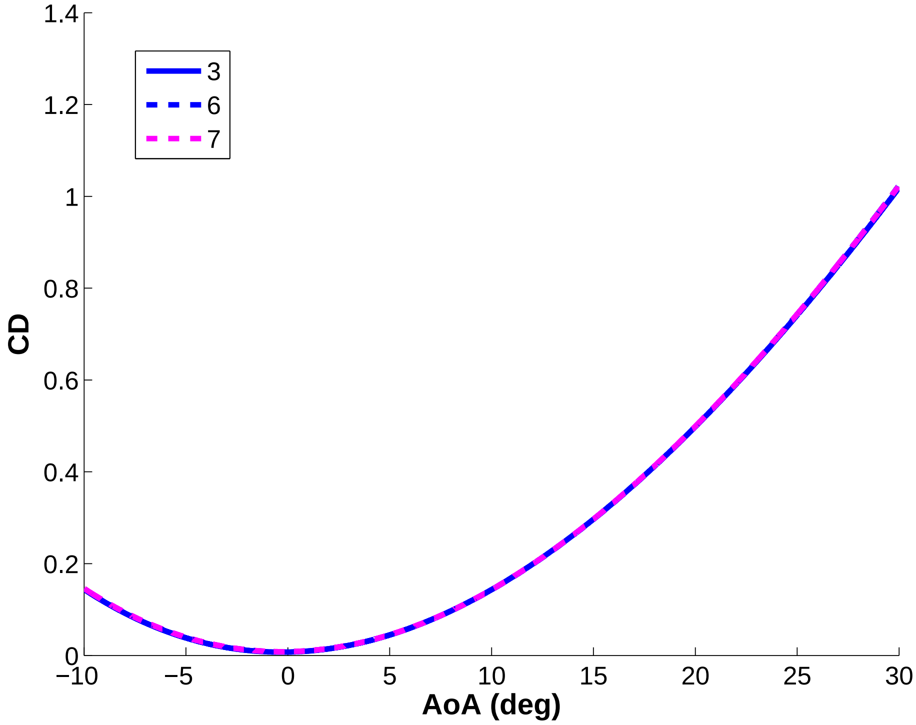


Figure 4.27: Comparison of all models predicting static C_D versus AoA curve.

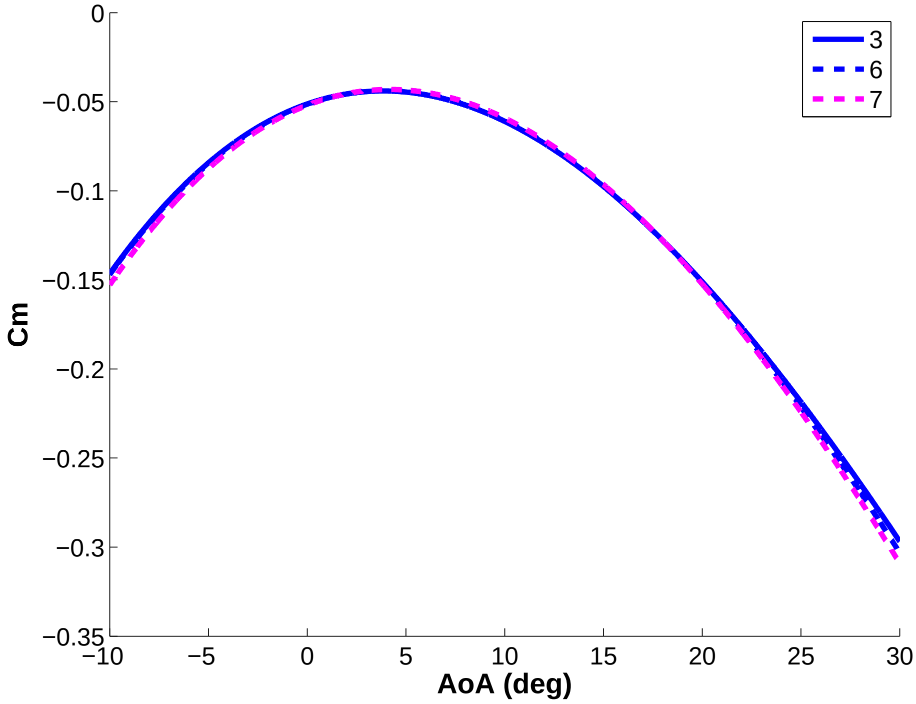


Figure 4.28: Comparison of all models predicting static C_m versus AoA curve.

coefficient than the currently best TM. This, of course, needs to be verified with the completion of the final jobs before this theory can be stated with conviction.

V. Conclusions

5.1 Summary

Inability to accurately predict nonlinear aerodynamics of high-performance aircraft early in the design cycle has been the source of many costly fixes to fighter aircraft after flight testing begins. Early prediction of these nonlinear effects will filter into many disciplines which must work together in the creation of an aircraft such as structures, performance, S&C, flight controls, and many others. CFD is a relatively new tool available to engineers and advances in modeling techniques and computational resources have advanced this field to a degree so as to provide high-fidelity results even in these nonlinear regions.

In this research, the theory of CFD and its proper use to obtain high-fidelity results was discussed in detail as well as the theory behind the why specific solver settings should be used and in what conditions. Focus on the importance of S&C parameters was also focused on as they describe the static and dynamic characteristics of the aircraft that need to be understood in order to filter the proper information into the other engineering disciplines mentioned. These characteristics of the aircraft are excited by the large-amplitude motions included herein and then modeled with SID techniques which have been utilized for flight and wind tunnel test results for years. This process of utilizing CFD to create the input and output data needed for SID techniques has been examined before, and the current state-of-the-art has been covered as well as its known downfalls.

The goal of this research was to help advance the current state-of-the-art by creating a set of metrics to describe the coverage of a TM over a desired regressor space which can be used to predict how well a TM will model that same regressor space. These metrics need to be validated before they can fulfill this last goal and this validation is done by creating a variety of TMs which are run and compared against validation data. The metrics which the best TMs excel would then be considered the most important to optimize when creating different TMs.

This goal was ultimately unattainable due to the lack of variation of being able to define a “best” TM. Each of the models predicted results similar to each other, including the current best TM. The only variation seen among the models is in the prediction of pitching moment coefficient where the TMs with static data were able to more closely model the pitch rate data from the CMs. TMs 3, 4, 5, and 6 had the best predictions of pitching moment coefficient but the metrics which are optimized by these four TMs varies from metrics 5 - 10: number of points at zero pitch rate and low pitch rate, the percentage of cells at zero pitch rate and low pitch rate with data points, and the “evenness” or standard deviation of the data across the same regions. It was found that the definition of “low pitch rate” should be no larger than 20 $\frac{deg}{s}$ as that is the region all four of the TMs discussed here fall within.

Some theories as to the lack of variability of the model predictions include:

- The possibility of too much linear regions in the regressor space.
- The time required for the dynamic transients to dampen out when the pitch rate is zero is too long and thus adversely affecting the results of the static regions of the model - pushing the values closer to the dynamic values.
- The time length of the TMs may have been too long - allowing for even the “worst” maneuvers to produce an accurate model.

A significant conclusion from these results concerns the trade-off between the addition of static data into a TM versus obtaining greater coverage of the regressor space through dynamic data only. It was found that the predictability of the dynamic results of a TM was not significantly degraded by the addition of the static regions. As a result, the importance of the static regions is verified even though the reason is different than originally hypothesized. The accurate prediction of the pitching moment data is most likely due to the unsteadiness introduced to pitching moment at high pitch rates, even at benign AoA values.

5.2 Recommendations

The completion of the cases not finished to date is of primary importance. Comparing the Mach 0.5 data to the static CM (CM3) may give extra insight into the ability of the chopped sinusoidal type motions to predict static data. Also, the completion of the Mach 0.9 cases may provide more insight into these TMs as transonic flow is inherently nonlinear and part of the reason why the Mach 0.5 models were so similar is that the regressor space might have included too large of a linear region.

If the completion of the current TMs does not produce a distinction among those used herein, then a different, much more nonlinear regressor space should be examined to see if a difference can be induced. If not, then the conclusions stated would be the only standing conclusions.

A potentially useful item to examine is the possibility of creating two models for the S&C data from the CFD results. The first one would be identical to the current process of passing the data through a low-pass filter and utilize the filtered data as the basis for determining the coefficients for the models. Then, look at the frequency of the fluctuations versus AoA by passing the data through a high-pass filter and create a model to describe these fluctuations – probably best done in the frequency domain. This would allow for the first model to predict the mean value and the second model predict the unsteady fluctuations.

5.2.1 Future Work. The first recommendation for future work is to utilize the new version of the Kestrel solver, due to be released in March, 2012. The current Kestrel versions (2.1.2) and prior include a version of kAVUS that suffers from various issues. These issues are isolated to Jacobian terms on the left-hand side, boundary conditions, and the turbulence models. Steady state accuracy is not affected by these issues, but they do cause problems with unsteady solution convergence depending on the quality of the mesh and the flow conditions. This is especially true for moving-grid problems and results in unsteady solutions having an undesirable dependency on temporal damping magnitudes. Version 2.2 of Kestrel corrects these deficiencies and future research should include an appropriate re-evaluation of the cases presented here with the updated version of Kestrel along with the utilization of AMR instead of the refinement based on a region used herein.

After conclusions are solidified about which TM is best, this TM can then be translated into three axes. After the TM is translated into three axes, this TM can be run at a wide range of Alt-Mach combinations to get a map of the characteristics of the aircraft over the flight envelope. These models can then be "fused" together into one overall ROM.

Another area of concern realized in the course of this work is the need for a standard method for grid and time convergence studies for moving grids. This process may be similar to the poor man's "steepest descent" method mentioned by Cummings, et al. for grid and time convergence in static simulations [13].

Last of all, this research held the length of time constant for each TM. Research may be done to determination of the minimum computational time needed to create a model to a given fidelity. This may also have been a reason of why the models results were similar – the regressor space may have been covered "too well" in the sense that all TMs were then able to produce accurate models whereas if the length of time was decreased there may have been more of a spread on how well each TM compared to the results.

Appendix A. ***Matlab®*** *Code for Metric Calculation*

Listing A.1: `Matlab`® Code Used to Calculate Metrics for all Training Maneuvers

```
function [metric metrica metricq as qs z]=Metric...
    (pos2,rate2,bounds,steps,metricsteps,maxN0,pm0)
aextrema=bounds(1,:);qextrema=bounds(2,:);
astep=steps(1);qstep=steps(2);
ametstep=metricsteps(1);qmetstep=metricsteps(2);
arange=aextrema(1):astep:aextrema(2);
qrange=qextrema(1):qstep:qextrema(2);
[as,qs]=meshgrid(...
    mean(arange(1:2)):astep:mean(arange(length(arange)-1:length(...
       arange))),...
    mean(qrange(1:2)):qstep:mean(qrange(length(qrange)-1:length(...
       qrange))));
z=zeros(size(as));
pos3=zeros(1,length(pos2)-1);
for i=2:length(pos2)
    pos3(i-1)=mean(pos2(i-1:i));
end
for i=1:length(rate2)
    if rate2(i)>qextrema(1) && rate2(i)<qextrema(2);
        for j=1:length(arange)-1
            apos=j;
            if pos3(i)<=arange(j+1)
                break
            end
        end
        for j=1:length(qrange)-1
            qpos=j;
            if rate2(i)<=qrange(j+1)
                break
            end
        end
        z(qpos,apos)=z(qpos,apos)+1;
    end
end
asteps=size(z,2)/ametstep;
counta=zeros(asteps,1);
totala=zeros(asteps,1);
totala(1)=size(z,1)*ametstep;
for i=1:size(z,1)
    k=1;
    for j=1:size(z,2)
        if z(i,j)~=0
            counta(k)=counta(k)+1;
        end
        if mod(j,ametstep)==0
            totala(k)=totala(1);
            k=k+1;
        end
    end
end
```

```
metrica=counta./totala;
% metric(1)=100*sum(counta)/sum(totala);

qsteps=size(z,1)/qmetstep;
countq=zeros(qsteps,1);
totalq=zeros(qsteps,1);
totalq(1)=size(z,2)*qmetstep;
for i=1:size(z,2)
    k=1;
    for j=1:size(z,1)
        if z(j,i)~=0
            countq(k)=countq(k)+1;
        end
        if mod(j,qmetstep)==0
            totalq(k)=totalq(1);
            k=k+1;
        end
    end
end
metricq=countq./totalq;
metric(1)=100*sum(countq)/sum(totalq);

count=0;
total=0;
j=size(z,2);
for i=1:size(z,1)
    if z(i,1)~=0
        count=count+1;
    end
    total=total+1;
    if z(i,j)~=0
        count=count+1;
    end
    total=total+1;
end
j=size(z,1);
for i=2:size(z,2)-1
    if z(1,i)~=0
        count=count+1;
    end
    total=total+1;
    if z(j,i)~=0
        count=count+1;
    end
    total=total+1;
end

for i=1:length(qrange)-1
    zeroqpos=i;
    if 0<=qrange(i+1)
        break
    end
```

```
end
metric(2)=100*count/total;
metric(3:4)=[std(metrica/max(metrica)) std(metricq/max(metricq))];

zsteps=size(z,2)/qmetstep;
countz=zeros(zsteps,1);
k=1;
i=zeroqpos;
for j=1:size(z,2)
    if z(i,j)~=0
        countz(k)=countz(k)+z(i,j);
    end
    if mod(j,qmetstep)==0
        k=k+1;
    end
end
% disp(countz')
countz12=0;
for i=1:length(countz)
    if countz(i)>maxN0
        countz(i)=maxN0;
    end
    if countz(i)>0
        countz12=countz12+1;
    end
end
% disp(countz')
metric(5)=100*countz12/length(countz);
metric(6)=mean(countz);
metric(7)=std(countz/max(countz));
% figure;plot(1:length(countz),countz)

countz22=0;
for mi=1:length(pm0)
    % countz2=z(zeroqpos-pm0(mi):zeroqpos+pm0(mi),:);
    countz2=[z(zeroqpos-pm0(mi):zeroqpos-1,:);z(zeroqpos+1:...
       zeroqpos+pm0(mi),:)];
    % disp(countz2)
    for i=1:size(countz2,1)
        for j=1:size(countz2,2)
            if countz2(i,j)>maxN0
                countz2(i,j)=maxN0;
            end
            if countz2(i,j)>0
                countz22=countz22+1;
            end
        end
    end
    % disp(countz2)
    metric(6+2*mi)=100*countz22/(size(countz2,1)*size(countz2,2));
    metric(7+2*mi)=mean(mean(countz2));
    for i=1:size(countz2,1)
```

```
        met8(i)=std(countz2(i,:)/max(countz2(i,:)));
    end
    j=1;
    for i=1:length(met8)
        if isnan(met8(i))
            met82(i)=0;
        else
            met82(i)=met8(i);
            %         met82(j)=met8(i);
            %         j=j+1;
        end
    end
    metric(8+2*mi)=mean(met82);
end

mll=9+2*mi;
countz3=[z(1:zeroqpos-pm0(mi),:);z(zeroqpos+pm0(mi):size(z,1),:)];
countz32=0;
for i=1:size(countz3,1)
    for j=1:size(countz3,2)
        if countz3(i,j)>0
            countz32=countz32+1;
        end
    end
end
metric(mll)=100*countz32/(size(countz3,1)*size(countz3,2));
metric(mll+1)=mean(mean(countz3));
for i=1:size(countz3,1)
    met10(i)=std(countz3(i,:)/max(countz3(i,:)));
end
j=1;
for i=1:length(met10)
    if isnan(met10(i))
        met102(i)=0;
    else
        met102(i)=met10(i);
%         met102(j)=met10(i);
%         j=j+1;
    end
end
metric(mll+2)=mean(met102);

%{
function [metric as qs z]=Metric(pos2,rate2,bounds,steps)




aextrema=bounds(1,:);qextrema=bounds(2,:);
astep=steps(1);qstep=steps(2);
arange=aextrema(1):astep:aextrema(2);
qrange=qextrema(1):qstep:qextrema(2);
```

```
[as,qs]=meshgrid(...
    mean(arange(1:2)):astep:mean(arange(length(arange)-1:length(...
       arange))),...
    mean(qrange(1:2)):qstep:mean(qrange(length(qrange)-1:length(...
       qrange))));
z=zeros(size(as));
for i=2:length(pos2)
    pos3(i-1)=mean(pos2(i-1:i));
end
for i=1:length(rate2)
    if rate2(i)>qextrema(1) && rate2(i)<qextrema(2)
        for j=1:length(arange)-1
            apos=j;
            if pos3(i)<=arange(j+1)
                break
            end
        end
        for j=1:length(qrange)-1
            qpos=j;
            if rate2(i)<=qrange(j+1)
                break
            end
        end
        z(qpos,apos)=z(qpos,apos)+1;
    end
end
count=0;
total=0;
for i=1:size(z,1)
    for j=1:size(z,2)
        if z(i,j)~=0
            count=count+1;
        end
        total=total+1;
    end
end
metric(1)=count/total;
count=0;
total=0;
j=size(z,2);
for i=1:size(z,1)
    if z(i,1)~=0
        count=count+1;
    end
    total=total+1;
    if z(i,j)~=0
        count=count+1;
    end
    total=total+1;
end
j=size(z,1);
for i=2:size(z,2)-1
```

```
        if z(1,i)~=0
            count=count+1;
        end
        total=total+1;
        if z(j,i)~=0
            count=count+1;
        end
        total=total+1;
    end
    metric(2)=count/total;
    %}

    end
```

Appendix B. Complete List of Solver Settings and Boundary Conditions

Table B.1: kAVUS Inputs

Input Parameter	(Static) & Motion Settings
Equation Set	Turbulent N-S
Inviscid Flux	Gottlieb and Groth
Turbulence Model	DDES
Turbulence Wall	No
Spatial Accuracy	Second-Order
Temporal Accuracy	Second-Order
Fixed Sweeps	No
Max Sweeps	64
Sweeps Convergence Criteria	1.0e-8
Temporal Damping Coeff (Adv)	Initial:0.6, Final:0.1-0.4
Temporal Damping Coeff (Diff)	1/10 of Temporal Damping Coeff (Adv)
Subiterations	5
Matrix Scheme	Gauss-Seidel
Limiter Type	Venkatakrishnan w/ K=2.5
Least Squares Type	Weighted
Theta	1.0
Gradient Type	Non-conservative
Stencil Type	Original
Wall Accuracy	Use spatial accuracy
Enable Gravity	No
Relaxation	0.7
Solution Update	Limited
Solution Average	No
Time Stepping Scheme	Global Specified

Listing B.1: Boundary Conditions for CFD Simulations

```
##################################################
  Boundary Condition Specification File for:
  F-16 Case
##################################################
 4
Fuselage
Solid-Wall
Adiabatic No Slip
yes
##################################################
 5
Wing
Solid-Wall
```

```
Adiabatic No Slip
yes
##################################################
 6
Canopy
Solid-Wall
Adiabatic No Slip
yes
##################################################
 7
Bump
Solid-Wall
Adiabatic No Slip
yes
##################################################
 10
Vertical Tail
Solid-Wall
Adiabatic No Slip
yes
##################################################
 11
Horizontal Tail
Solid-Wall
Adiabatic No Slip
yes
##################################################
 12
Fin
Solid-Wall
Adiabatic No Slip
yes
##################################################
 20
Inlet
Solid-Wall
Adiabatic No Slip
yes
##################################################
 30
Engine Exhaust
Source
Riemann
 P-Stat  T-Stat  K or Nu~/Nu  Omega   Mach    Axis End Points     ...
    Swirl (A,B,C)
  25.00   2500.     -1.          -1.    1.2     0. 0. 0.  1. 0. 0. ...
         0. 0. 0.
No
##################################################
 21
Engine Face
Sink
```

```
Mass Flow
 Mass Flow
   0.6578800 1.0
No
##################################################
 22
Diverter Face
Solid-Wall
Slip
No
##################################################
 3
Outer Boundary
Farfield
Modified Riemann Invariants
 P-Stat    T-Stat   K or Nu~/Nu  Omega    Mach    Alpha    Beta
  -1.       -1.      -1.           -1.    -1.     -370.    -370.
##################################################
```

Listing C.1: Matlab® Code Used to Calculate Training Maneuvers

```
close all;clear all;
% %{
out='Y';wrt=10;whichplots=[2 3];
t0=0;tmax=5;dt=.0002;
t=t0:dt:tmax;
that=t-t0;

aoai=8;

limit=1000;
bounds=[-10 30;-100 100];
steps=[.1;.10]*1;
% steps=[.1;.1];
metricsteps=[1;1];
maxN0=1000/dt;pm0=[20/steps(2)];% 10/steps(2) 20/steps(2) 50/steps...
   (2)];
figcount=0;

%% 1st maneuver
% %{
a1=22.5; a2=0; f1=.8; f2=f1; lamf=1; phi=-90;
bf=(f2-f1)/(tmax-t0)^lamf;
s1=(a1:(a2-a1)/(length(that)-1):a2);
trad2=s1.*cos(2*pi.*(bf*that.^(1+lamf)+f1*that+phi/360));
trad2=trad2+aoai;

ratet2=zeros(length(trad2)-1,1);
accelt2=zeros(length(trad2)-2,1);
for i=2:length(trad2)
    ratet2(i-1)=(trad2(i)-trad2(i-1))/dt;
end
for i=2:length(ratet2)
    accelt2(i-1)=(ratet2(i)-ratet2(i-1))/dt;
end

[metric2 metrica2 metricq2 as2 qs2 z2]=Metric...
    (trad2,ratet2,bounds,steps,metricsteps,maxN0,pm0);
text1=num2str(pm0(1)*steps(2));
% disp('                                            pts_@_0Q ...
              pts +/- 5deg/s      pts +/- 10deg/s     pts +/- 20...
   deg/s     pts +/- 50deg/s     pts_high_Q')
% disp('    Whole     Boundary  A_std     Q_std     per_cell  ...
   std_dev   per_cell  std_dev   per_cell  std_dev   per_cell  ...
   std_dev   per_cell  std_dev   per_cell  std_dev')
disp(['                                            ...
   __________pts_@_0Q_________   ______pts +/- ' text1 ' deg/...
   s_____   _________pts_high_Q________'])
disp( '    Whole     Boundary  A_std     Q_std     percent   ...
   per_cell  std_dev   percent   per_cell  std_dev   percent   ...
   per_cell  std_dev')
disp(['1:  ' num2str(metric2, '%10.6f')])

% z2=zeros(20,40);
if out=='Y'
    [figcount]=Plotdata(that,trad2,ratet2,accelt2,metric2,as2,qs2,...
       z2,whichplots,figcount,1);
end

if wrt==1 || wrt==10
    Write_Motion(that,trad2,aoai,'man1.mtn');
end

%}

%% 2nd maneuver
% %{
a1=22.5; a2=0; f1=.75; f2=1.6; lamf=1.9; phi=-90;
bf=(f2-f1)/(tmax-t0)^lamf;
s1=(a1:(a2-a1)/(length(that)-1):a2);
trad3=s1.*cos(2*pi.*(bf*that.^(1+lamf)+f1*that+phi/360));
trad3=trad3+aoai;

ratet3=zeros(length(trad3)-1,1);
accelt3=zeros(length(trad3)-2,1);
for i=2:length(trad3)
    ratet3(i-1)=(trad3(i)-trad3(i-1))/dt;
end
for i=2:length(ratet3)
    accelt3(i-1)=(ratet3(i)-ratet3(i-1))/dt;
end

[metric3 metrica3 metricq3 as3 qs3 z3]=Metric...
    (trad3,ratet3,bounds,steps,metricsteps,maxN0,pm0);
% disp('Maneuver 2:                                pts_@_0Q ...
              pts_around_0Q')
% disp('    Whole     Boundary  A_std     Q_std     per_cell   ...
   std_dev   per_cell  std_dev')
% disp(metric3)
disp(['2:  ' num2str(metric3, '%10.6f')])

if out=='Y'
    [figcount]=Plotdata(that,trad3,ratet3,accelt3,metric3,as3,qs3,...
       z3,whichplots,figcount,1);
end

if wrt==2 || wrt==10
    Write_Motion(that,trad3,aoai,'man2.mtn');
end

%}

%% 3rd maneuver
```

```
% %{
a1=27; a2=0; f1=.6; f2=1.35; lamf=1.9; phi=-90;
bf=(f2-f1)/(tmax-t0)^lamf;
s1=(a1:(a2-a1)/(length(that)-1):a2);
pos=s1.*cos(2*pi.*(bf*that.^(1+lamf)+f1*that+phi/360));
pos=pos+aoai;

extrema=[30 -10 23 -5 18 0 15 2.5 12 5 10 6.5 8.5 8 8 8];
mark=1;
j=length(pos)+1;
k=1;
for i=1:length(pos)
    if mark == 1
        if pos(i)> extrema(k)
            pos(i)= extrema(k);
            j = i;
        end
    end
    if mark == 2
        if pos(i)< extrema(k)
            pos(i)= extrema(k);
            j = i;
        end
    end
    if i>j
        j=length(pos)+1;
        k=k+1;
        if mark == 1
            mark=2;
        else
            mark=1;
        end
        if k > length(extrema)
            break
        end
    end
end
rate=zeros(length(pos)-1,1);
accel=zeros(length(pos)-2,1);
for i=2:length(pos)
    rate(i-1)=(pos(i)-pos(i-1))/dt;
end
for i=2:length(rate)
    accel(i-1)=(rate(i)-rate(i-1))/dt;
end

[pos2 rate2 accel2]=accel_smooth(pos,rate,accel,dt,limit);
% pos2=pos;rate2=rate;accel2=accel;
[metric1 metrica1 metricq1 as1 qs1 z1]=Metric...
    (pos2,rate2,bounds,steps,metricsteps,maxN0,pm0);
% disp('Maneuver 3:                                    pts_@_0Q ...
              pts_around_0Q')
```

```
% disp('    Whole     Boundary  A_std     Q_std     per_cell  ...
   std_dev   per_cell  std_dev')
% disp(metric1)
disp(['3:  ' num2str(metric1, '%10.6f')])

if out=='Y'
    [figcount]=Plotdata(that,pos2,rate2,accel,metric1,as1,qs1,z1,...
       whichplots,figcount,1);
end

if wrt==3 || wrt==10
    Write_Motion(that,pos2,aoai,'man3.mtn');
end
%}

%% 4th maneuver
% %{
cut=round(4*.005/dt);
a1=27; a2=0; f1=.6; f2=1.35; lamf=1.9; phi=-90;
bf=(f2-f1)/(tmax-t0)^lamf;
s1=(a1:(a2-a1)/(length(that)-1):a2);
pos=s1.*cos(2*pi.*(bf*that.^(1+lamf)+f1*that+phi/360));
pos=pos+aoai;

extrema=[30 -10 23 -5 18 0 15 2.5 12 5 10 6.5 8.5 8 8 8];

mark=1;
j=length(pos)+1;
k=1;
for i=1:length(pos)
    if mark == 1
        if pos(i)> extrema(k)
            pos(i)= extrema(k);
            j = i;
        end
    end
    if mark == 2
        if pos(i)< extrema(k)
            pos(i)= extrema(k);
            j = i;
        end
    end
    if i>j
        j=length(pos)+1;
        k=k+1;
        if mark == 1
            mark=2;
        else
            mark=1;
        end
        if k > length(extrema)
            break
        end
    end
end
rate4=zeros(length(pos)-1,1);
accel4=zeros(length(pos)-2,1);

that2=t0:cut*dt:tmax;pos22=pos(1:cut:length(pos));
spl = spline(that2,pos22,that);
pp = spline(that2,pos22);
pos4=spl;
for i=2:length(pos4);rate4(i-1)=(pos4(i)-pos4(i-1))/dt;end
for i=2:length(rate4);accel4(i-1)=(rate4(i)-rate4(i-1))/dt;end
[metric4 metrica4 metricq4 as4 qs4 z4]=Metric...
    (pos4,rate4,bounds,steps,metricsteps,maxN0,pm0);
% disp('Maneuver 4:                                 pts_@_0Q ...
              pts_around_0Q')
% disp('    Whole     Boundary  A_std     Q_std     per_cell  ...
   std_dev   per_cell  std_dev')
% disp(metric4)
disp(['4:  ' num2str(metric4, '%10.6f')])

if out=='Y'
    [figcount]=Plotdata(that,pos4,rate4,accel4,metric4,as4,qs4,z4,...
       whichplots,figcount,1);
end

if wrt==4 || wrt==10
    Write_Motion(that,pos4,aoai,'man4.mtn');
end

%}


%% 5th maneuver
% %{
cut=round(9*.005/dt);
a1=27; a2=0; f1=.6; f2=1.35; lamf=1.9; phi=-90;
bf=(f2-f1)/(tmax-t0)^lamf;
s1=(a1:(a2-a1)/(length(that)-1):a2);
pos=s1.*cos(2*pi.*(bf*that.^(1+lamf)+f1*that+phi/360));
pos=pos+aoai;

extrema=[30 -10 23 -5 18 0 15 2.5 12 5 10 6.5 8.5 8 8 8];

mark=1;
j=length(pos)+1;
k=1;
for i=1:length(pos)
    if mark == 1
        if pos(i)> extrema(k)
            pos(i)= extrema(k);
            j = i;
        end
    end
    if mark == 2
        if pos(i)< extrema(k)
            pos(i)= extrema(k);
            j = i;
        end
    end
    if i>j
        j=length(pos)+1;
        k=k+1;
        if mark == 1
            mark=2;
        else
            mark=1;
        end
        if k > length(extrema)
            break
        end
    end
end
rate5=zeros(length(pos)-1,1);
accel5=zeros(length(pos)-2,1);

that2=t0:cut*dt:tmax;pos22=pos(1:cut:length(pos));
spl = spline(that2,pos22,that);
pos5=spl;
for i=2:length(pos5);rate5(i-1)=(pos5(i)-pos5(i-1))/dt;end
for i=2:length(rate5);accel5(i-1)=(rate5(i)-rate5(i-1))/dt;end
[metric5 metrica5 metricq5 as5 qs5 z5]=Metric...
    (pos5,rate5,bounds,steps,metricsteps,maxN0,pm0);
% disp('Maneuver 5:                                 pts_@_0Q ...
              pts_around_0Q')
% disp('    Whole     Boundary  A_std     Q_std     per_cell  ...
   std_dev   per_cell  std_dev')
% disp(metric5)
disp(['5:  ' num2str(metric5, '%10.6f')])

if out=='Y'
    [figcount]=Plotdata(that,pos5,rate5,accel5,metric5,as5,qs5,z5,...
       whichplots,figcount,1);
end

if wrt==5 || wrt==10
    Write_Motion(that,pos5,aoai,'man5.mtn');
end

%}

%% 6th maneuver
% %{
cut=round(3*.005/dt);qmax=20;
```

```
a1=27; a2=0; f1=.6; f2=1.35; lamf=1.9; phi=-90;
bf=(f2-f1)/(tmax-t0)^lamf;
s1=(a1:(a2-a1)/(length(that)-1):a2);
pos=s1.*cos(2*pi.*(bf*that.^(1+lamf)+f1*that+phi/360));
pos=pos+aoai;

extrema=[29.75 -9.75 23 -5 18 0 15 2.5 12 5 10 6.5 8.5 8 8 8];
mark=1;
j=length(pos)+1;
k=1;
for i=1:length(pos)
    if mark == 1
        if pos(i)> extrema(k)
            pos(i)= extrema(k);
            j = i;
        end
    end
    if mark == 2
        if pos(i)< extrema(k)
            pos(i)= extrema(k);
            j = i;
        end
    end
    if i>j
        j=length(pos)+1;
        k=k+1;
        if mark == 1
            mark=2;
        else
            mark=1;
        end
        if k > length(extrema)
            break
        end
    end
end
rate=zeros(length(pos)-1,1);
accel6=zeros(length(pos)-2,1);
for i=2:length(pos)
    rate(i-1)=(pos(i)-pos(i-1))/dt;
end

[pos6 rate6 accel6]=accel_smooth2(pos,rate,accel,dt,qmax);
that2=t0:cut*dt:tmax;pos26=pos6(1:cut:length(pos6));
pos6 = spline(that2,pos26,that);
for i=2:length(rate6);accel6(i-1)=(rate6(i)-rate6(i-1))/dt;end
[metric6 metrica6 metricq6 as6 qs6 z6]=Metric...
    (pos6,rate6,bounds,steps,metricsteps,maxN0,pm0);
% disp('Maneuver 6:                                   pts_@_0Q ...
              pts_around_0Q')
% disp('    Whole     Boundary  A_std     Q_std     per_cell  ...
   std_dev   per_cell  std_dev')
```

```
% disp(metric6)
disp(['6:   ' num2str(metric6, '%10.6f')])

if out=='Y'
    [figcount]=Plotdata(that,pos6,rate6,accel6,metric6,as6,qs6,z6,...
        whichplots,figcount,1);
end

if wrt==6 || wrt==10
    Write_Motion(that,pos6,aoai,'man6.mtn');
end
%}

%% 7th maneuver
% %{
cut=round(3*.005/dt);qmax=50;
a1=27; a2=0; f1=.6; f2=1.35; lamf=1.9; phi=-90;
bf=(f2-f1)/(tmax-t0)^lamf;
s1=(a1:(a2-a1)/(length(that)-1):a2);
pos=s1.*cos(2*pi.*(bf*that.^(1+lamf)+f1*that+phi/360));
pos=pos+aoai;

extrema=[29 -9 23 -5 18 0 15 2.5 12 5 10 6.5 8.5 8 8 8];
mark=1;
j=length(pos)+1;
k=1;
for i=1:length(pos)
    if mark == 1
        if pos(i)> extrema(k)
            pos(i)= extrema(k);
            j = i;
        end
    end
    if mark == 2
        if pos(i)< extrema(k)
            pos(i)= extrema(k);
            j = i;
        end
    end
    if i>j
        j=length(pos)+1;
        k=k+1;
        if mark == 1
            mark=2;
        else
            mark=1;
        end
        if k > length(extrema)
            break
        end
    end
end
```

```
rate=zeros(length(pos)-1,1);
accel7=zeros(length(pos)-2,1);
for i=2:length(pos)
    rate(i-1)=(pos(i)-pos(i-1))/dt;
end

[pos7 rate7 accel7]=accel_smooth2(pos,rate,accel,dt,qmax);
that2=t0:cut*dt:tmax;pos27=pos7(1:cut:length(pos7));
pos7 = spline(that2,pos27,that);
for i=2:length(rate7);accel7(i-1)=(rate7(i)-rate7(i-1))/dt;end
[metric7 metrica7 metricq7 as7 qs7 z7]=Metric...
    (pos7,rate7,bounds,steps,metricsteps,maxN0,pm0);
% disp('Maneuver 7:                              pts_@_0Q ...
              pts_around_0Q')
% disp('    Whole     Boundary  A_std     Q_std     per_cell  ...
   std_dev   per_cell  std_dev')
% disp(metric7)
disp(['7:   ' num2str(metric7, '%10.6f')])
% out='Y';
if out=='Y'
    [figcount]=Plotdata(that,pos7,rate7,accel7,metric7,as7,qs7,z7,...
       whichplots,figcount,1);
end

if wrt==7 || wrt==10
    Write_Motion(that,pos7,aoai,'man7.mtn');
end
%}
```

Listing C.2: Matlab® Code Used to Calculate Comparison Maneuvers

```
close all;clear all;
% %{
out='Y';wrt=0;whichplots=[2,3];
t0=0;tmax=5;dt=.0002;
t=t0:dt:tmax;
that=t-t0;

aoai=8;

limit=1000;
bounds=[-10 30;-100 100];
steps=[.1;.1];metricsteps=[1;1];
maxN0=1000/dt;pm0=[20/steps(2)];% 10/steps(2) 20/steps(2) 50/steps...
   (2)];
figcount=0;

%% 1st maneuver
% %{
cut=round(5*.005/dt);
a1=0; a2=24; f1=1.8; f2=1.2; lamf=1.2; phi=90;
bf=(f2-f1)/(tmax-t0)^lamf;
s1=(a1:(a2-a1)/(length(that)-1):a2);
```

```
pos=s1.*cos(2*pi.*(bf*that.^(1+lamf)+f1*that+phi/360));
aoai=8;
pos=pos+aoai;

% extrema=[30 -10 25 -5 20 -2.5 15 0 12 2.5 10 5 8 6 7 7 7];
extrema=[-10 30 -5 18 0 15 2.5 12 5 10 6.5 8.5 8 8 8];

mark=2;
j=length(pos)+1;
k=1;
for i=1:length(pos)
    if mark == 1
        if pos(i)> extrema(k)
            pos(i)= extrema(k);
            j = i;
        end
    end
    if mark == 2
        if pos(i)< extrema(k)
            pos(i)= extrema(k);
            j = i;
        end
    end
    if i>j
        j=length(pos)+1;
        k=k+1;
        if mark == 1
            mark=2;
        else
            mark=1;
        end
        if k > length(extrema)
            break
        end
    end
end
rate=zeros(length(pos)-1,1);
accel=zeros(length(pos)-2,1);
for i=2:length(pos)
    rate(i-1)=(pos(i)-pos(i-1))/dt;
end
for i=2:length(rate)
    accel(i-1)=(rate(i)-rate(i-1))/dt;
end

that2=t0:cut*dt:tmax;pos22=pos(1:cut:length(pos));
spl = spline(that2,pos22,that);
pos2=spl;
for i=2:length(pos2);rate2(i-1)=(pos2(i)-pos2(i-1))/dt;end
accel2=accel;

[metric1 metrica1 metricq1 as1 qs1 z1]=Metric...
```

```
    (pos2,rate2,bounds,steps,metricsteps,maxN0,pm0);
text1=num2str(pm0(1)*steps(2));
disp(['                                                 ...
   __________pts_@_0Q_________    ______pts +/- ' text1 ' deg/...
   s_____   _________pts_high_Q________'])
disp( '    Whole     Boundary  A_std     Q_std     percent   ...
   per_cell  std_dev   percent   per_cell  std_dev   percent   ...
   per_cell  std_dev')
disp(['1:   ' num2str(metric1, '%10.4f')])

if out=='Y'
    [figcount]=Plotdata(that,pos2,rate2,[],metric1,as1,qs1,z1,...
       whichplots,figcount,1);
end

if wrt==1 || wrt==10
    Write_Motion(that,pos2,aoai,'Comp1.mtn');
end

%}

%% 2nd maneuver
% %{
t0=0;tmax=2.5;
t=t0:dt:tmax;
that=t-t0;

a1=1; a2=4; f1=.6; f2=.8; lamf=1; phi=-90;
bf=(f2-f1)/(tmax-t0)^lamf;
s1=(a1:(a2-a1)/(length(that)-1):a2);
comp=s1.*cos(2*pi.*(bf*that.^(1+lamf)+f1*that+phi/360));

that=0:dt:5;
half=length(comp);
inter=[comp comp(2:half)];

for i=1:half-1
    inter(half+i)=-comp(half-i);
end

trad2=inter+aoai;

ratet2=zeros(length(trad2)-1,1);
accelt2=zeros(length(trad2)-2,1);
for i=2:length(trad2)
    ratet2(i-1)=(trad2(i)-trad2(i-1))/dt;
end
for i=2:length(ratet2)
    accelt2(i-1)=(ratet2(i)-ratet2(i-1))/dt;
end

```

```
[metric2 metrica2 metricq2 as2 qs2 z2]=Metric...
    (trad2,ratet2,bounds,steps,metricsteps,maxN0,pm0);
% disp('Comparison Maneuver 2:                      pts_@_0Q ...
              pts_around_0Q')
% disp('    Whole     Boundary  A_std     Q_std     per_cell  ...
   std_dev   per_cell  std_dev')
disp(['2:   ' num2str(metric2, '%10.4f')])

if out=='Y'
    [figcount]=Plotdata(that,trad2,ratet2,[],metric2,as2,qs2,z2,...
       whichplots,figcount,2);
end

if wrt==1 || wrt==10
    Write_Motion(that,trad2,aoai,'Comp2.mtn');
end

%}

%% 3rd maneuver
%{

[coeff,flow,motion] = readtrackoutput('F16');

trad2 = motion.AOA;

ratet2=motion.Q;

% for i=2:length(ratet2)
%     accelt2(i-1)=(ratet2(i)-ratet2(i-1))/dt;
% end

%%

% [metric2 metrica2 metricq2 as2 qs2 z2]=Metric...
%     (trad2,ratet2,bounds,steps,metricsteps,maxN0,pm0);
% disp('Comparison Maneuver 2:                      pts_@_0Q ...
              pts_around_0Q')
% disp('    Whole     Boundary  A_std     Q_std     per_cell  ...
   std_dev   per_cell  std_dev')
% disp(['2:   ' num2str(metric2, '%10.4f')])
that = motion.TIME;
for i=2:length(trad2)
    rate2(i-1)=(trad2(i)-trad2(i-1))/dt;
end
if out=='Y'
    [figcount]=Plotdata(that,trad2,rate2,[],[],[],[],[],whichplots...
       ,figcount,1);
end

```

```
%}
```

Listing C.3: `Matlab`® Code Used to Round Corners of TM3

```
function [pos2 rate2 accel2]=accel_smooth(pos,rate,accel,dt,limit)
pos2=pos;
rate2=rate;
accel2=accel;
i=0;
while i<length(accel)
    i=i+1;
    j=i+2;
    if accel(i)>limit
        t1=pos(i); t2=pos(j);
        if t2<t1
            x1=i-1;
            y1=pos(i-1);
            y2=pos(j+1);
            s1=(pos(i)-pos(i-2))/2;
            x0=x1+(y2-y1)*s1/(sqrt(s1^2+1)-1);
            y0=y1+(x1-x0)/s1;
            r=y0-y2;
            pos2(x1:floor(x0))=-sqrt(r^2-([x1:floor(x0)]-x0).^2)+...
                y0;
            i=floor(x0);
        else
            x1=j+2;
            y1=pos(j+2);
            y2=pos(i);
            s1=(pos(j+3)-pos(j+1))/2;
            x0=x1+(y2-y1)*s1/(sqrt(s1^2+1)-1);
            y0=y1+(x1-x0)/s1;
            r=y2-y0;
            pos2(ceil(x0):x1)=-sqrt(r^2-([ceil(x0):x1]-x0).^2)+y0;
            i=x1;
        end
    elseif accel(i)<-limit
        t1=pos(i); t2=pos(j);
        if t2<t1
            x1=j+2;
            y1=pos(j+2);
            y2=pos(i);
            s1=(pos(j+3)-pos(j+1))/2;
            x0=x1+(y2-y1)*s1/(sqrt(s1^2+1)-1);
            y0=y1+(x1-x0)/s1;
            r=y2-y0;
            pos2(ceil(x0):x1)=sqrt(r^2-([ceil(x0):x1]-x0).^2)+y0;
            i=x1;
        else
            x1=i-1;
            y1=pos(i-1);
            y2=pos(j+1);
            s1=(pos(i)-pos(i-2))/2;
            x0=x1+(y2-y1)*s1/(sqrt(s1^2+1)-1);
            y0=y1+(x1-x0)/s1;
            r=y2-y0;
            pos2(x1:floor(x0))=sqrt(r^2-([x1:floor(x0)]-x0).^2)+y0...
               ;
            i=floor(x0);
        end
    end
end

for i=2:length(pos)
    rate2(i-1)=(pos2(i)-pos2(i-1))/dt;
end
for i=2:length(rate)
    accel2(i-1)=(rate2(i)-rate2(i-1))/dt;
end
end
```

Listing C.4: `Matlab`® Code Used to Produce Low Pitch Rate Oscillations for TM6 and TM7

```
function [pos2 rate2 accel2]=accel_smooth2(pos,rate,accel,dt,qmax)
pos2=pos;
rate2=rate;
accel2=accel;
i=0;
while i<length(pos)-1
    i=i+1;
    if pos(i+1)==pos(i)
        mark1=i;
        for j=i:length(pos)
            if pos(j)~=pos(mark1) || j==length(pos)
                mark2=j-1;
                break
            end
        end
        i=mark2+2;
        T=dt*(mark2-mark1)/1.5;
        w=2*pi/T;
        A=qmax/(w);
        if pos(mark1)<pos(mark1-1)
            A=-A;
        end
        for k=mark1:mark2
            pos2(k)=A*sin(w*(k-mark1)*dt)+pos(mark1);
        end
    end
```

```
end

for i=2:length(pos)
    rate2(i-1)=(pos2(i)-pos2(i-1))/dt;
end
for i=2:length(rate)
    accel2(i-1)=(rate2(i)-rate2(i-1))/dt;
end
end
```

Listing C.5: `Matlab`® Code Used to Create Plots of all Maneuvers

```
function [i]=Plotdata(that,pos2,rate2,accel2,metric1,as1,qs1,z1,...
   whichplots,i,choi)

% h=figure;set(h,'color',[1 1 1]);colormap(bone);patch(X,Y,-x');
% axis equal;axis tight;axis off
% print('-dpng','I:\My Documents\Thesis\ThesisDoc\Paint\P1mv2.png...
   ')
% print('-deps','I:\My Documents\Thesis\ThesisDoc\Figures\eps\ ...
   P1mv2.eps')

fsize=16;
asize=14;
fweight='b';
ii=(i/length(whichplots))+1;
whichplots=[whichplots 0];
j=1;
if whichplots(j)==1
    j=j+1;
    i=i+1;
    h=figure(i);set(h,'color',[1 1 1]);
    [AX,H1,H2] = plotyy(that,pos2,that(2:length(that)),rate2);%,...
       that(3:length(that)),accel2)
    xlabel('Time (s)','fontsize',fsize,'fontweight',fweight);
    set(H1,'LineStyle','-','linewidth',2)
    set(H2,'LineStyle','--','linewidth',2)
    set(get(AX(1),'Ylabel'),'String','AoA (deg)','fontsize',fsize,...
       'fontweight',fweight)
    set(get(AX(2),'Ylabel'),'String','Pitch Rate (Q)','fontsize',...
       fsize,'fontweight',fweight)
    if choi==1
        set(AX(1),'YLim',[-12 32],'YTick',[-10:5:30])
        set(AX(2),'YLim',[-120 120],'YTick',[-100:25:100])
    end
    name=['Time_AoAQ_' num2str(ii) '.eps'];
    print('-depsc',name)
end
if whichplots(j)==2
    j=j+1;
    i=i+1;
    h=figure(i);set(h,'color',[1 1 1]);
    plot(pos2(2:length(pos2)),rate2,'.','markersize',10)
    xlabel('AoA (deg)','fontsize',fsize,'fontweight',fweight);
    ylabel('Pitch Rate (Q)','fontsize',fsize,'fontweight',fweight)...
       ;
    set(gca,'fontsize',asize)
    if choi==1
        set(gca,'XTick',-10:5:30)
        set(gca,'YTick',-100:25:100)
        axis([-12 32 -120 120]);
    end
%     title({['Total %= ' num2str(metric1(1)*100)...
%          '  Boundary %= ' num2str(metric1(2)*100)];...
%          ['Std Dev AoA= ' num2str(metric1(3))...
%          '  Std Dev Q= ' num2str(metric1(4))];...
%          ['# pts per cell @ 0 Q= ' num2str(metric1(5))...
%          '  Std Dev Pts @ 0 Q= ' num2str(metric1(6))];...
%          ['# pts per cell ~ 0 Q= ' num2str(metric1(7))...
%          '  Std Dev Pts ~ 0 Q= ' num2str(metric1(8))];
%          ['# pts per cell highQ= ' num2str(metric1(9))...
%          '  Std Dev Pts highQ= ' num2str(metric1(10))];})
    name=['AoA_Q_' num2str(ii) '.eps'];
    print('-depsc',name)
end
if whichplots(j)==3
    j=j+1;
    i=i+1;
    h=figure(i);set(h,'color',[1 1 1]);
    plot(that,pos2,'b','linewidth',2)
    set(gca,'fontsize',asize)
    set(gca,'XTick',0:1:5)
    if choi==1
        set(gca,'YTick',-10:5:30)
        axis([0 5 -12 32]);
    end
    xlabel('Time (s)','fontsize',fsize,'fontweight',fweight);
    ylabel('AoA (deg)','fontsize',fsize,'fontweight',fweight);
    name=['Time_AoA_' num2str(ii) '.eps'];
    print('-depsc',name)
end
if whichplots(j)==4
    j=j+1;
    i=i+1;
    h=figure(i);surf(as1,qs1,z1);
    set(h,'color',[1 1 1]);
    set(gca,'fontsize',asize)
    view(gca,[-30 45]);
%     grid(axes1,'on');
%     hold(axes1,'all');
    xlabel('AoA','fontsize',fsize,'fontweight',fweight);
    ylabel('Pitch Rate (Q)','fontsize',fsize,'fontweight',fweight)...
       ;
    zlabel('Number of Occurances','fontsize',fsize,'fontweight',...
       fweight)
```

```
%     title({['Total %= ' num2str(metric1(1)*100)...
%         '  Boundary %= ' num2str(metric1(2)*100)];...
%         ['Std Dev AoA= ' num2str(metric1(3))...
%         '  Std Dev Q= ' num2str(metric1(4))];...
%         ['# pts per cell @ 0 Q= ' num2str(metric1(5))...
%         '  Std Dev Pts @ 0 Q= ' num2str(metric1(6))];...
%         ['# pts per cell ~ 0 Q= ' num2str(metric1(7))...
%         '  Std Dev Pts ~ 0 Q= ' num2str(metric1(8))];
%         ['# pts per cell highQ= ' num2str(metric1(9))...
%         '  Std Dev Pts highQ= ' num2str(metric1(10))];})
%     print('-dpng','Test1.png')
    name=['Regressor_' num2str(ii) '.eps'];
    print('-depsc',name)

end

end
```

Listing D.1: `Matlab`® Code Used to Produce Arbitrary Motion File for Kestrel Input

```
function [] = Write_Motion(that,pos2,aoai,filename)
%% Write motion to file for Kestrel input
if isempty(filename)
     filename='ChopSinArb.mtn';
end
fid = fopen(filename, 'w');
a1='####################################################';
a2='Arbitrary Motion File:';
a3='Insert Descriptive Title Here';
a4='######################################################';
a5='Motion Reference Frame (Body or Mesh):';
a6='Mesh';
a7='######################################################';
a8='Time Rotated Basis Vectors (x3,y3,z3) Cur. Position of Center ...
    of Rotation';
fprintf(fid, '%s \n', a1);
fprintf(fid, '%s \n', a2);
fprintf(fid, '%s \n', a3);
fprintf(fid, '%s \n', a4);
fprintf(fid, '%s \n', a5);
fprintf(fid, '%s \n', a6);
fprintf(fid, '%s \n', a7);
fprintf(fid, '%s \n', a8);

aoas=pos2-aoai;
for i=1:length(pos2)
    A=[that(i) cosd(aoas(i)) -sind(aoas(i)) 0 sind(aoas(i)) cosd(...
        aoas(i)) 0 0 0 1 320.654 91.0 0];
    fprintf(fid, ...
         '%8.6f %9.6f %9.6f %9.6f %9.6f %9.6f %9.6f %9.6f %9.6f ...
             %9.6f %9.6f %9.6f %9.6f \n',A);
end
fprintf(fid, '%s', 'end');
fclose(fid);
end
```

Listing E.1: `Matlab`® Code Used to Load Data for Analysis

```
%Load Data
close all;clear all;clc;

for i=1:7
    filename=[’Man’ num2str(i)];
    cd ..
    cd(filename)
    [iter(i,:),time(i,:),CL(i,:),CD(i,:),Cm(i,:),aoa(i,:),Q(i,:)] ...
       = readtrackoutput2(’F16’);    %#ok<*SAGROW>
end

for i=1:2
    filename=[’Comp’ num2str(i)];
    cd ..
    cd(filename)
    [iterC(i,:),timeC(i,:),CLC(i,:),CDC(i,:),CmC(i,:),aoaC(i,:),QC...
       (i,:)] = readtrackoutput2(’F16’);
end

filename=[’Comp’ num2str(3)];
cd ..
cd(filename)
[iterC3,timeC3,CLC3,CDC3,CmC3,aoaC3,QC3] = readtrackoutput2(’F16’)...
    ;

cd ../Models_max2

iterC(3,:)=zeros(1,size(iterC,2));
iterC(3,1:length(iterC3))=iterC3;

% timeC(3,:)=zeros(1,size(timeC,2));
% timeC(3,1:length(timeC3))=timeC3;
timeC(3,:)=timeC(2,:);

CLC(3,:)=zeros(1,size(CLC,2));
CLC(3,1:length(CLC3))=CLC3;

CDC(3,:)=zeros(1,size(CDC,2));
CDC(3,1:length(CDC3))=CDC3;

CmC(3,:)=zeros(1,size(CmC,2));
CmC(3,1:length(CmC3))=CmC3;

aoaC(3,:)=zeros(1,size(aoaC,2));
aoaC(3,1:length(aoaC3))=aoaC3;

QC(3,:)=zeros(1,size(QC,2));
QC(3,1:length(QC3))=QC3;

% clear iterC3 timeC3 CLC3 CDC3 CmC3 aoaC3 QC3 filename i
```

Listing E.2: Matlab® Code Used to Read Tracking Data Outputs from Kestrel

```
function [iter,time,CL,CD,Cm,aoa,Q] = readtrackoutput2(filename)
% Imports Kestrel tracking output files
%   "filename" is the file name w/o the extensions
fid1 = fopen([filename '.coeff']);
fid2 = fopen([filename '.motion']);
for skip = 1:20
    fgetl(fid1);
    fgetl(fid2);
end
%i = 0;
%while ~feof(fid1) % check for end of file
    %i = i + 1;
    A = fscanf(fid1,' %f %f %f %f %f %f %f %f %f %f %f %f ',[12,...
       inf]);
%     for i = 1:size(A,2)
        iter    = A(1,:);
        time    = A(2,:);
%         aoa     = A(3,:);
%         coeff.BETA    = A(4,:);
%         coeff.CAXIAL  = A(5,:);
%         coeff.CNORMAL = A(6,:);
        CL   = A(7,:);
        CD   = A(8,:);
%         coeff.CSIDE   = A(9,:);
        Cm  = A(10,:);
%         coeff.CROLL   = A(11,:);
%         coeff.CYAW    = A(12,:);
%     end
%     A;
clear A

%i = 0;
%while ~feof(fid2) % check for end of file
    %i = i + 1;
    B = fscanf(fid2,' %f %f %f %f %f %f %f %f %f %f %f %f %f %f %f...
        %f %f %f %f %f %f %f %f %f %f',[25,inf]);
%     for i = 1:size(B,2)
%         flow.ITER    = B(1,:);
%         flow.TIME    = B(2,:);
        aoa = B(3,:);
%         flow.NONLINP = B(4,:);
%         flow.YPLUS   = B(5,:);
%         flow.SUPER   = B(6,:);
%         flow.FX      = B(7,:);
%         flow.FY      = B(8,:);
%         flow.FZ      = B(9,:);
%         flow.MX      = B(10,:);
%         flow.MY      = B(11,:);
        Q   = B(12,:);
%     end
clear B
```

```
end
```

Listing E.3: `Matlab`® Code Used to Create Plots of Models

```
%PlotModels
function []=PlotModels(time,CM,CA,ylab,leg,rs)
lw=1;fsize=16;asize=14;fweight='b';

for i=1:m
    h=figure;set(h,'color',[1 1 1]);
    plot(time(i,:),CA(i,:),'k-','linewidth',lw)
    hold on
    plot(time(i,:),CM(i,:),'r:','linewidth',lw)
    xlabel('Time (s)','fontsize',fsize,'fontweight',fweight);
    ylabel(ylab,'fontsize',fsize,'fontweight',fweight);
    set(gca,'fontsize',asize)
    legend('Actual',leg)%,'fontsize',asize)
    title(['R^2= ' num2str(rs(i)*100)],'fontsize',asize)
%     name=[name1 num2str(i) '.eps'];
%     print('-depsc',name)
end


end
```

Listing E.4: `Matlab`® Code Used to Analyze Output Models

```
%Compare Models
close all;clc;clear CD1 ybar St Sr CL1 Cm1 CL2 CL3 CL4 CL5 CL6 CL7...
    ...
CD2 CD3 CD4 CD5 CD6 CD7 Cm2 Cm3 Cm4 Cm5 Cm6 Cm7
%Must first run 'LoadData.m' for this code to work!!!!!!

n=2;m=2;
%Model from Maneuver 1:
for i=n:m
    CD1(i,:)=(9.789573e-03)+(3.336984e-04).*aoaC(i,:).^2+(1.529731...
       e-05).*...
        aoaC(i,:).*QC(i,:)-(1.396869e-04).*aoaC(i,:)-(2.072500e...
           -05).*...
        QC(i,:)+(1.196021e-07).*QC(i,:).^2;
    ybar=mean(CDC(i,:));
    St=sum((CDC(i,:)-ybar).^2);
    Sr=sum((CD1(i,:)-CDC(i,:)).^2);
    rsCD1(i)=abs((St-Sr)/St);

    CL1(i,:)=2.235669e-02+2.100859e-02.*aoaC(i,:)+6.996580e...
       -04.*...
        QC(i,:)-1.175294e-04.*aoaC(i,:).^2+6.192553e-06...
        .*aoaC(i,:).*QC(i,:)+4.882452e-07.*QC(i,:).^2;
    ybar=mean(CLC(i,:));
    St=sum((CLC(i,:)-ybar).^2);
    Sr=sum((CL1(i,:)-CLC(i,:)).^2);
    rsCL1(i)=abs((St-Sr)/St);

    Cm1(i,:)=-1.681662e-02-5.707444e-04.*QC(i,:)+2.590801e-03.*...
        aoaC(i,:)-1.062944e-04.*aoaC(i,:).^2+5.133947e-06.*aoaC(i...
           ,:)...
        .*QC(i,:)+6.123563e-08.*QC(i,:).^2; %#ok<*SAGROW>
    ybar=mean(CmC(i,:));
    St=sum((CmC(i,:)-ybar).^2);
    Sr=sum((Cm1(i,:)-CmC(i,:)).^2);
    rsCm1(i)=abs((St-Sr)/St);
end

% PlotModels(timeC,CD1,CDC,'CD','Model1',rsCD1)
% PlotModels(timeC,CL1,CLC,'CL','Model1',rsCL1)
% PlotModels(timeC,Cm1,CmC,'Cm','Model1',rsCm1)

%Model from Maneuver 2:
for i=n:m
    CD2(i,:)=9.971521e-03+3.345139e-04.*aoaC(i,:).^2+1.528962e...
       -05.*...
        aoaC(i,:).*QC(i,:)-1.858821e-05.*QC(i,:)-1.461004e-04.*...
           aoaC(i,:);
    ybar=mean(CDC(i,:));
    St=sum((CDC(i,:)-ybar).^2);
    Sr=sum((CD2(i,:)-CDC(i,:)).^2);
    rsCD2(i)=abs((St-Sr)/St);

    CL2(i,:)=2.307955e-02+2.107215e-02.*aoaC(i,:)+6.957933e-04.*QC...
       (i,:)...
        -1.177817e-04.*aoaC(i,:).^2+8.516905e-06.*aoaC(i,:).*QC(i...
           ,:)-...
        1.087788e-07.*QC(i,:).^2;
    ybar=mean(CLC(i,:));
    St=sum((CLC(i,:)-ybar).^2);
    Sr=sum((CL2(i,:)-CLC(i,:)).^2);
    rsCL2(i)=abs((St-Sr)/St);

    Cm2(i,:)=-1.679564e-02-5.632180e-04.*QC(i,:)+2.504913e-03.*...
        aoaC(i,:)-9.978461e-05.*aoaC(i,:).^2+5.130502e-06.*...
        aoaC(i,:).*QC(i,:)-1.081843e-07.*QC(i,:).^2;
    ybar=mean(CmC(i,:));
    St=sum((CmC(i,:)-ybar).^2);
    Sr=sum((Cm2(i,:)-CmC(i,:)).^2);
    rsCm2(i)=abs((St-Sr)/St);
end

% PlotModels(timeC,CD2,CDC,'CD','Model2',rsCD2)
% PlotModels(timeC,CL2,CLC,'CL','Model2',rsCL2)
% PlotModels(timeC,Cm2,CmC,'Cm','Model2',rsCm2)

%Model from Maneuver 3:
for i=n:m
```

```
    CD3(i,:)=9.912525e-03+3.256221e-04.*aoaC(i,:).^2+1.393598e...
       -05.*...
        aoaC(i,:).*QC(i,:)-1.437804e-05.*QC(i,:)+1.468528e-07.*...
        QC(i,:).^2-4.635287e-05.*aoaC(i,:);
    ybar=mean(CDC(i,:));
    St=sum((CDC(i,:)-ybar).^2);
    Sr=sum((CD3(i,:)-CDC(i,:)).^2);
    rsCD3(i)=abs((St-Sr)/St);

    CL3(i,:)=2.387139e-02+2.097867e-02.*aoaC(i,:)+7.156790e...
       -04.*...
        QC(i,:)-1.252798e-04.*aoaC(i,:).^2+5.070804e-07.*QC(i,:)...
           .^2+...
        2.015581e-06.*aoaC(i,:).*QC(i,:);
    ybar=mean(CLC(i,:));
    St=sum((CLC(i,:)-ybar).^2);
    Sr=sum((CL3(i,:)-CLC(i,:)).^2);
    rsCL3(i)=abs((St-Sr)/St);

    Cm3(i,:)=-1.790076e-02-5.588308e-04.*QC(i,:)+2.457503e-03.*...
        aoaC(i,:)-8.962253e-05.*aoaC(i,:).^2+2.155569e-07.*QC(i,:)...
           .^2+...
        4.550380e-07.*aoaC(i,:).*QC(i,:);
    ybar=mean(CLC(i,:));
    St=sum((CmC(i,:)-ybar).^2);
    Sr=sum((Cm3(i,:)-CmC(i,:)).^2);
    rsCm3(i)=abs((St-Sr)/St);
end

% PlotModels(timeC,CD3,CDC,’CD’,’Model3’,rsCD3)
% PlotModels(timeC,CL3,CLC,’CL’,’Model3’,rsCL3)
% PlotModels(timeC,Cm3,CmC,’Cm’,’Model3’,rsCm3)

%Model from Maneuver 4:
for i=n:m
    CD4(i,:)=1.007479e-02+3.262006e-04.*aoaC(i,:).^2+1.465288e...
       -05.*...
        aoaC(i,:).*QC(i,:)-1.693442e-05.*QC(i,:)-5.606118e-05.*...
        aoaC(i,:)+8.137841e-08.*QC(i,:).^2;
    ybar=mean(CDC(i,:));
    St=sum((CDC(i,:)-ybar).^2);
    Sr=sum((CD4(i,:)-CDC(i,:)).^2);
    rsCD4(i)=abs((St-Sr)/St);

    CL4(i,:)=2.452260e-02+2.093719e-02.*aoaC(i,:)+7.182334e...
       -04.*...
        QC(i,:)-1.245912e-04.*aoaC(i,:).^2+4.062658e-06.*aoaC(i,:)...
           .*...
        QC(i,:)+2.545718e-07.*QC(i,:).^2;
    ybar=mean(CLC(i,:));
    St=sum((CLC(i,:)-ybar).^2);
    Sr=sum((CL4(i,:)-CLC(i,:)).^2);
    rsCL4(i)=abs((St-Sr)/St);

    Cm4(i,:)=-1.703279e-02-5.572417e-04.*QC(i,:)+2.435494e-03.*...
        aoaC(i,:)-9.418785e-05.*aoaC(i,:).^2+2.347739e-06.*aoaC(i...
           ,:).*...
        QC(i,:)+6.924145e-08.*QC(i,:).^2;
    ybar=mean(CLC(i,:));
    St=sum((CmC(i,:)-ybar).^2);
    Sr=sum((Cm4(i,:)-CmC(i,:)).^2);
    rsCm4(i)=abs((St-Sr)/St);
end

% PlotModels(timeC,CD4,CDC,'CD','Model4',rsCD4)
% PlotModels(timeC,CL4,CLC,'CL','Model4',rsCL4)
% PlotModels(timeC,Cm4,CmC,'Cm','Model4',rsCm4)


%Model from Maneuver 5:
for i=n:m
    CD5(i,:)=9.993058e-03+3.269691e-04.*aoaC(i,:).^2+1.436076e...
       -05.*...
        aoaC(i,:).*QC(i,:)-1.623005e-05.*QC(i,:)+9.284830e-08.*...
        QC(i,:).^2-5.130136e-05.*aoaC(i,:);
    ybar=mean(CDC(i,:));
    St=sum((CDC(i,:)-ybar).^2);
    Sr=sum((CD5(i,:)-CDC(i,:)).^2);
    rsCD5(i)=abs((St-Sr)/St);

    CL5(i,:)=2.450979e-02+2.092801e-02.*aoaC(i,:)+7.227792e...
       -04.*...
        QC(i,:)-1.222738e-04.*aoaC(i,:).^2+3.069282e-06.*aoaC(i,:)...
           .*...
        QC(i,:)+2.955129e-07.*QC(i,:).^2;
    ybar=mean(CLC(i,:));
    St=sum((CLC(i,:)-ybar).^2);
    Sr=sum((CL5(i,:)-CLC(i,:)).^2);
    rsCL5(i)=abs((St-Sr)/St);

    Cm5(i,:)=-1.687697e-02-5.616563e-04.*QC(i,:)+2.455266e-03.*...
        aoaC(i,:)-9.542033e-05.*aoaC(i,:).^2+2.940546e-06.*aoaC(i...
           ,:).*...
        QC(i,:)+4.286884e-08.*QC(i,:).^2;
    ybar=mean(CLC(i,:));
    St=sum((CmC(i,:)-ybar).^2);
    Sr=sum((Cm5(i,:)-CmC(i,:)).^2);
    rsCm5(i)=abs((St-Sr)/St);
end

% PlotModels(timeC,CD5,CDC,'CD','Model5',rsCD5)
% PlotModels(timeC,CL5,CLC,'CL','Model5',rsCL5)
% PlotModels(timeC,Cm5,CmC,'Cm','Model5',rsCm5)
```

```
%Model from Maneuver 6:
for i=n:m
    CD6(i,:)=1.003183e-02+3.275261e-04.*aoaC(i,:).^2+1.456518e...
       -05.*...
        aoaC(i,:).*QC(i,:)-1.593029e-05.*QC(i,:)-6.653847e-05.*...
        aoaC(i,:)+6.521430e-08.*QC(i,:).^2;
    ybar=mean(CDC(i,:));
    St=sum((CDC(i,:)-ybar).^2);
    Sr=sum((CD6(i,:)-CDC(i,:)).^2);
    rsCD6(i)=abs((St-Sr)/St);

    CL6(i,:)=2.451389e-02+2.091614e-02.*aoaC(i,:)+7.215477e...
       -04.*...
        QC(i,:)-1.223068e-04.*aoaC(i,:).^2+4.044164e-06.*aoaC(i,:)...
           .*...
        QC(i,:)+1.867449e-07.*QC(i,:).^2;
    ybar=mean(CLC(i,:));
    St=sum((CLC(i,:)-ybar).^2);
    Sr=sum((CL6(i,:)-CLC(i,:)).^2);
    rsCL6(i)=abs((St-Sr)/St);

    Cm6(i,:)=-1.701866e-02-5.607149e-04.*QC(i,:)+2.452204e-03.*...
        aoaC(i,:)-9.580407e-05.*aoaC(i,:).^2+3.169494e-06.*...
        aoaC(i,:).*QC(i,:)+5.142742e-08.*QC(i,:).^2;
    ybar=mean(CLC(i,:));
    St=sum((CmC(i,:)-ybar).^2);
    Sr=sum((Cm6(i,:)-CmC(i,:)).^2);
    rsCm6(i)=abs((St-Sr)/St);
end

% PlotModels(timeC,CD6,CDC,'CD','Model6',rsCD6)
% PlotModels(timeC,CL6,CLC,'CL','Model6',rsCL6)
% PlotModels(timeC,Cm6,CmC,'Cm','Model6',rsCm6)


%Model from Maneuver 7:
for i=n:m
    CD7(i,:)=9.958869e-03+3.300937e-04.*aoaC(i,:).^2+1.377499e...
       -05.*...
        aoaC(i,:).*QC(i,:)-1.153540e-05.*QC(i,:)-8.913833e-05.*...
        aoaC(i,:)+5.712025e-08.*QC(i,:).^2;
    ybar=mean(CDC(i,:));
    St=sum((CDC(i,:)-ybar).^2);
    Sr=sum((CD7(i,:)-CDC(i,:)).^2);
    rsCD7(i)=abs((St-Sr)/St);

    CL7(i,:)=2.365951e-02+2.103502e-02.*aoaC(i,:)+7.270375e...
       -04.*...
        QC(i,:)-1.224904e-04.*aoaC(i,:).^2+2.165246e-06.*aoaC(i,:)...
           .*...
        QC(i,:)+1.354950e-07.*QC(i,:).^2;
    ybar=mean(CLC(i,:));
    St=sum((CLC(i,:)-ybar).^2);
    Sr=sum((CL7(i,:)-CLC(i,:)).^2);
    rsCL7(i)=abs((St-Sr)/St);

    Cm7(i,:)=-1.709275e-02-5.585180e-04.*QC(i,:)+2.463586e-03.*...
        aoaC(i,:)-9.600510e-05.*aoaC(i,:).^2+3.277661e-06.*...
        aoaC(i,:).*QC(i,:);
    ybar=mean(CLC(i,:));
    St=sum((CmC(i,:)-ybar).^2);
    Sr=sum((Cm7(i,:)-CmC(i,:)).^2);
    rsCm7(i)=abs((St-Sr)/St);
end

% PlotModels(timeC,CD7,CDC,'CD','Model7',rsCD7)
% PlotModels(timeC,CL7,CLC,'CL','Model7',rsCL7)
% PlotModels(timeC,Cm7,CmC,'Cm','Model7',rsCm7)

lw=3;fsize=16;asize=14;fweight='b';
for i=n:m
    h=figure;set(h,'color',[1 1 1]);
    plot(timeC(i,:),CLC(i,:),'k-','linewidth',4)
    hold on
    plot(time(i,:),CL1(i,:),'r-','linewidth',lw)
    plot(time(i,:),CL2(i,:),'g-','linewidth',lw)
    plot(time(i,:),CL3(i,:),'b-','linewidth',lw)
    plot(time(i,:),CL4(i,:),'m-','linewidth',lw)
    plot(time(i,:),CL5(i,:),'g--','linewidth',lw)
    plot(time(i,:),CL6(i,:),'b--','linewidth',lw)
    plot(time(i,:),CL7(i,:),'m--','linewidth',lw)
    set(gca,'fontsize',asize)
    xlabel('Time (s)','fontsize',fsize,'fontweight',fweight);
    ylabel('CL','fontsize',fsize,'fontweight',fweight);
    legend('Comp','1','2','3','4','5','6','7')
    title(['R^{2}_{1}= ' num2str(rsCL1(i)),', R^{2}_{2} ' num2str(...
       rsCL2(i)),', R^{2}_{3} ' ...
        num2str(rsCL3(i)),', R^{2}_{4} ' num2str(rsCL4(i)),', R...
           ^{2}_{5} ' ...
        num2str(rsCL5(i)),', R^{2}_{6} ' num2str(rsCL6(i)),', R...
           ^{2}_{7} ' num2str(rsCL7(i))])
end

for i=n:m
    h=figure;set(h,'color',[1 1 1]);
    plot(timeC(i,:),CDC(i,:),'k-','linewidth',4)
    hold on
    plot(time(i,:),CD1(i,:),'r-','linewidth',lw)
    plot(time(i,:),CD2(i,:),'g-','linewidth',lw)
    plot(time(i,:),CD3(i,:),'b-','linewidth',lw)
    plot(time(i,:),CD4(i,:),'m-','linewidth',lw)
    plot(time(i,:),CD5(i,:),'g--','linewidth',lw)
    plot(time(i,:),CD6(i,:),'b--','linewidth',lw)
    plot(time(i,:),CD7(i,:),'m--','linewidth',lw)
    set(gca,'fontsize',asize)
    xlabel('Time (s)','fontsize',fsize,'fontweight',fweight);
    ylabel('CD','fontsize',fsize,'fontweight',fweight);
    legend('Comp','1','2','3','4','5','6','7')
        title(['R^{2}_{1}' num2str(rsCD1(i)),', R^{2}_{2}' num2str...
           (rsCD2(i)),', R^{2}_{3}' ...
        num2str(rsCD3(i)),', R^{2}_{4}' num2str(rsCD4(i)),', R^{2}...
           _{5}' num2str(rsCD5(i)),', R^{2}_{6}'...
        num2str(rsCD6(i)),', R^{2}_{7}' num2str(rsCD7(i))])
end

for i=n:m
    h=figure;set(h,'color',[1 1 1]);
    plot(timeC(i,:),CmC(i,:),'k-','linewidth',4)
    hold on
    plot(time(i,:),Cm1(i,:),'r-','linewidth',lw)
    plot(time(i,:),Cm2(i,:),'g-','linewidth',lw)
    plot(time(i,:),Cm3(i,:),'b-','linewidth',lw)
    plot(time(i,:),Cm4(i,:),'m-','linewidth',lw)
    plot(time(i,:),Cm5(i,:),'g--','linewidth',lw)
    plot(time(i,:),Cm6(i,:),'b--','linewidth',lw)
    plot(time(i,:),Cm7(i,:),'m--','linewidth',lw)
    set(gca,'fontsize',asize)
    xlabel('Time (s)','fontsize',fsize,'fontweight',fweight);
    ylabel('Cm','fontsize',fsize,'fontweight',fweight);
    legend('Comp','1','2','3','4','5','6','7')
        title(['R^{2}_{1}' num2str(rsCm1(i)),', R^{2}_{2}' num2str...
           (rsCm2(i)),', R^{2}_{3}' ...
        num2str(rsCm3(i)),', R^{2}_{4}' num2str(rsCm4(i)),', R^{2}...
           _{5}' num2str(rsCm5(i)),', R^{2}_{6}'...
        num2str(rsCm6(i)),', R^{2}_{7}' num2str(rsCm7(i))])
end

% print('-depsc','CL_Comp1_M5_max2.eps')


%% Static Data Analysis

Astat=-10:.01:30;
Qstat=zeros(1,length(Astat))+0;

%Model from Maneuver 1:
    CD1s=9.730597e-03+3.431586e-04.*Astat.^2+1.114767e-05.*...
        Astat.*Qstat-1.863392e-04.*Astat+2.175502e-07.*...
        Astat.^2.*Qstat-2.179208e-05.*Qstat-3.443936e-07.*...
        Astat.^3+1.909637e-08.*Astat.*Qstat.^2-8.172904e-08.*...
        Qstat.^2+7.794505e-10.*Qstat.^3;
    CL1s=1.716054e-02+2.081710e-02.*Astat+6.690807e-04.*...
        Qstat-4.813678e-06.*Astat.^3+7.158772e-07.*Astat.^2.*...
        Qstat-7.668215e-06.*Astat.*Qstat+7.978169e-09.*...
        Qstat.^3+1.583122e-05.*Astat.^2+2.569745e-07.*...
        Qstat.^2+7.375897e-09.*Astat.*Qstat.^2;
    Cm1s=-2.046480e-02-5.460982e-04.*Qstat+2.417664e-03.*Astat...
       -3.625706e-06.*Astat.^3+3.044896e-07.*Astat.^2.*Qstat...
       -3.860851e-09.*Qstat.^3+2.844594e-08.*Astat.*Qstat...
       .^2-3.065744e-07.*Qstat.^2-6.108701e-06.*Astat.^2-4.261713e...
       -07.*Astat.*Qstat; %#ok<*SAGROW>

%Model from Maneuver 2:
    CD2s=9.319225e-03+3.503486e-04.*Astat.^2+1.225731e-05.*Astat.*...
       Qstat-1.703414e-04.*Astat+1.638292e-07.*Astat.^2.*Qstat...
       -5.657110e-07.*Astat.^3-7.683677e-06.*Qstat-8.013186e-10.*...
       Qstat.^3;
    CL2s=1.692528e-02+2.088811e-02.*Astat+7.252946e-04.*Qstat...
       -5.104902e-06.*Astat.^3+3.107823e-07.*Astat.^2.*Qstat...
       +2.495334e-05.*Astat.^2+2.766175e-06.*Astat.*Qstat-1.303175...
       e-08.*Astat.*Qstat.^2-2.810888e-09.*Qstat.^3;
    Cm2s=-2.163996e-02-5.333153e-04.*Qstat+2.478468e-03.*Astat...
       -3.378626e-06.*Astat.^3-4.521698e-08.*Astat.*Qstat...
       .^2+2.087651e-07.*Astat.^2.*Qstat+2.836222e-07.*Qstat...
       .^2-3.386342e-09.*Qstat.^3+1.283839e-06.*Astat.*Qstat...
       -5.663656e-06.*Astat.^2;

%Model from Maneuver 3:
    CD3s=9.252898e-03+3.516546e-04.*Astat.^2+1.284095e-05.*Astat.*...
       Qstat-8.600562e-07.*Astat.^3+3.505367e-08.*Astat.*Qstat...
       .^2-1.536454e-04.*Astat-1.679629e-07.*Qstat.^2+5.357569e...
       -08.*Astat.^2.*Qstat-1.131920e-09.*Qstat.^3-3.974588e-06.*...
       Qstat;
    CL3s=1.762809e-02+2.063312e-02.*Astat+7.435046e-04.*Qstat...
       -5.105015e-06.*Astat.^3+8.481360e-08.*Astat.*Qstat...
       .^2+2.769515e-05.*Astat.^2+9.290891e-08.*Astat.^2.*Qstat...
       -2.601925e-07.*Qstat.^2-3.163653e-09.*Qstat.^3;
    Cm3s=-1.946036e-02+2.299033e-03.*Astat-5.503736e-04.*Qstat...
       -1.597633e-06.*Astat.^3-4.146868e-05.*Astat.^2+4.770809e...
       -08.*Astat.*Qstat.^2+3.474431e-06.*Astat.*Qstat-1.587192e...
       -07.*Astat.^2.*Qstat-2.119689e-07.*Qstat.^2-1.746067e-09.*...
       Qstat.^3;

%Model from Maneuver 4:
    CD4s=9.209800e-03+3.512430e-04.*Astat.^2+1.269150e-05.*Astat.*...
       Qstat-8.323128e-07.*Astat.^3-1.313685e-04.*Astat+2.201535e...
       -08.*Astat.*Qstat.^2+9.902947e-08.*Astat.^2.*Qstat-1.170078...
       e-07.*Qstat.^2-8.455517e-06.*Qstat-6.842621e-10.*Qstat.^3;
    CL4s=1.724702e-02+2.070095e-02.*Astat+7.423128e-04.*Qstat...
       -5.190808e-06.*Astat.^3+2.040528e-07.*Astat.^2.*Qstat...
       +3.035673e-05.*Astat.^2+3.773327e-08.*Astat.*Qstat...
       .^2-2.312165e-09.*Qstat.^3-8.982990e-08.*Qstat.^2;
    Cm4s=-1.968283e-02+2.342294e-03.*Astat-5.374400e-04.*Qstat...
       -1.917533e-06.*Astat.^3-3.689443e-05.*Astat.^2+3.296173e...
       -06.*Astat.*Qstat+1.599009e-08.*Astat.*Qstat.^2-2.893298e...
       -09.*Qstat.^3-5.165568e-08.*Astat.^2.*Qstat-7.281210e-08.*...
       Qstat.^2;
```

```
%Model from Maneuver 5:
    CD5s=9.130721e-03+3.524162e-04.*Astat.^2+1.258388e-05.*Astat.*...
       Qstat-8.469415e-07.*Astat.^3-1.280549e-04.*Astat+2.327615e...
       -08.*Astat.*Qstat.^2-1.295475e-05.*Qstat+8.877916e-08.*...
       Astat.^2.*Qstat-1.207452e-07.*Qstat.^2;
    CL5s=1.696247e-02+2.069711e-02.*Astat+7.420088e-04.*Qstat...
       -5.378247e-06.*Astat.^3+3.785705e-05.*Astat.^2+3.831630e...
       -08.*Astat.*Qstat.^2+2.060893e-07.*Astat.^2.*Qstat-1.029270...
       e-06.*Astat.*Qstat-1.641683e-09.*Qstat.^3-7.006502e-08.*...
       Qstat.^2;
    Cm5s=-2.021508e-02+2.397853e-03.*Astat-5.422347e-04.*Qstat...
       -2.177549e-06.*Astat.^3-3.078932e-05.*Astat.^2+1.748001e...
       -06.*Astat.*Qstat-2.400336e-09.*Qstat.^3+6.050437e-08.*...
       Astat.^2.*Qstat+3.481904e-08.*Qstat.^2;

%Model from Maneuver 6:
    CD6s=9.241321e-03+3.502276e-04.*Astat.^2+1.287600e-05.*Astat.*...
       Qstat-7.555134e-07.*Astat.^3-1.336560e-04.*Astat-1.304469e...
       -05.*Qstat+8.551504e-08.*Astat.^2.*Qstat+1.804724e-08.*...
       Astat.*Qstat.^2-9.829171e-08.*Qstat.^2;
    CL6s=1.704703e-02+2.070347e-02.*Astat+7.287251e-04.*Qstat...
       -5.173246e-06.*Astat.^3+2.022190e-07.*Astat.^2.*Qstat...
       +3.201886e-05.*Astat.^2+2.089257e-08.*Astat.*Qstat.^2;
    Cm6s=-2.006787e-02+2.369637e-03.*Astat-5.433397e-04.*Qstat...
       -2.075239e-06.*Astat.^3-3.385414e-05.*Astat.^2+1.000226e...
       -07.*Astat.^2.*Qstat-1.993588e-09.*Qstat.^3+6.588607e-09.*...
       Astat.*Qstat.^2+1.182338e-06.*Astat.*Qstat;

%Model from Maneuver 7:
    CD7s=9.295198e-03+3.520622e-04.*Astat.^2+1.322779e-05.*Astat.*...
       Qstat-7.314771e-07.*Astat.^3-1.653406e-04.*Astat+1.578238e...
       -08.*Astat.*Qstat.^2-1.274319e-05.*Qstat-8.560719e-08.*...
       Qstat.^2+2.732796e-08.*Astat.^2.*Qstat+3.151080e-10.*Qstat...
       .^3;
    CL7s=1.682322e-02+2.078854e-02.*Astat+7.133831e-04.*Qstat...
       -4.984327e-06.*Astat.^3+2.623087e-05.*Astat.^2+1.543615e...
       -08.*Astat.*Qstat.^2+1.468376e-06.*Astat.*Qstat+2.227071e...
       -09.*Qstat.^3+3.621003e-08.*Astat.^2.*Qstat;
    Cm7s=-2.052570e-02-5.649622e-04.*Qstat+2.413161e-03.*Astat...
       -2.239398e-06.*Astat.^3-2.961360e-05.*Astat.^2+1.950899e...
       -07.*Astat.^2.*Qstat+1.711179e-09.*Qstat.^3-1.188499e-08.*...
       Astat.*Qstat.^2+8.199981e-08.*Qstat.^2-5.416685e-07.*Astat...
       .*Qstat;


    h=figure;set(h,'color',[1 1 1]);
    plot(Astat,CL1s,'r-','linewidth',lw)
    hold on
    plot(Astat,CL2s,'g-','linewidth',lw)
    plot(Astat,CL3s,'b-','linewidth',lw)
    plot(Astat,CL4s,'m-','linewidth',lw)
    plot(Astat,CL5s,’g--’,’linewidth’,lw)
    plot(Astat,CL6s,’b--’,’linewidth’,lw)
    plot(Astat,CL7s,’m--’,’linewidth’,lw)
    set(gca,’fontsize’,asize)
    xlabel(’AoA (deg)’,’fontsize’,fsize,’fontweight’,fweight);
    ylabel(’CL’,’fontsize’,fsize,’fontweight’,fweight);
    legend(’1’,’2’,’3’,’4’,’5’,’6’,’7’)

    h=figure;set(h,’color’,[1 1 1]);
    plot(Astat,CD1s,’r-’,’linewidth’,lw)
    hold on
    plot(Astat,CD2s,’g-’,’linewidth’,lw)
    plot(Astat,CD3s,’b-’,’linewidth’,lw)
    plot(Astat,CD4s,’m-’,’linewidth’,lw)
    plot(Astat,CD5s,’g--’,’linewidth’,lw)
    plot(Astat,CD6s,’b--’,’linewidth’,lw)
    plot(Astat,CD7s,’m--’,’linewidth’,lw)
    set(gca,’fontsize’,asize)
    xlabel(’AoA (deg)’,’fontsize’,fsize,’fontweight’,fweight);
    ylabel(’CD’,’fontsize’,fsize,’fontweight’,fweight);
    legend(’1’,’2’,’3’,’4’,’5’,’6’,’7’)

    h=figure;set(h,’color’,[1 1 1]);
    plot(Astat,Cm1s,’r-’,’linewidth’,lw)
    hold on
    plot(Astat,Cm2s,’g-’,’linewidth’,lw)
    plot(Astat,Cm3s,’b-’,’linewidth’,lw)
    plot(Astat,Cm4s,’m-’,’linewidth’,lw)
    plot(Astat,Cm5s,’g--’,’linewidth’,lw)
    plot(Astat,Cm6s,’b--’,’linewidth’,lw)
    plot(Astat,Cm7s,’m--’,’linewidth’,lw)
    set(gca,’fontsize’,asize)
    xlabel(’AoA (deg)’,’fontsize’,fsize,’fontweight’,fweight);
    ylabel(’Cm’,’fontsize’,fsize,’fontweight’,fweight);
    legend(’1’,’2’,’3’,’4’,’5’,’6’,’7’)

% print(’-depsc’,’Cm_Static2_M5_max2.eps’)
```

Bibliography

1. URL `http://www.centennialofflight.gov/essay/Theories_of_Flight/Stability/TH26G6.htm`. Centennial of Flight.
2. "Fixed-Wing Aerodynamics and Performance". URL `https://rdl.train.army.mil/soldierPortal/atia/adlsc/view/public/23947-1/FM/3-04.203/chap7.htm`. US Army.
3. "Theories of Flight". URL `http://www.centennialofflight.gov/essay/Theories_of_Flight/Two_dimensional_coef/TH14G4.htm`. U.S. Centennial of Flight Commission.
4. "SpinTraining", 2012. URL `http://flyacro.us/SpinTraining.html`. Aero Enterprises.
5. "Strake", 2012. URL `http://www.answers.com/topic/strake`. Reference Answers.
6. Anderson, John D. "Airfoil", 2008. URL `http://accessscience.com/content/Airfoil/019500`. AccessScience, McGraw-Hill Companies.
7. Anderson, John D. *Fundamentals of Aerodynamics*. McGraw-Hill, 5 edition, 2011.
8. Blazek, J. *Computational Fluid Dynamics: Principles and Applications*. Elsevier, 2 edition, 2005.
9. Bruce, P. D. and M. G. Kellet. "Modeling and Identification of Nonlinear Aerodynamic Functions Using B-Splines". *Journal of Aerospace Engineering*, 214(1/2000):27–40, 2000. Proceedings of the Institution of Mechanical Engineers, Part G.
10. Chaderjian, N. M., J. Ahmad, S. Pandya, and S. Murman. "Progress Toward Generation of a Navier-Stokes Database for a Harrier in Ground Effect". *AIAA*, 2002. 2002-5966.
11. Claus, M., S. A. Morton, and R. Cummings. "DES Turbulence Modelling on the C-130 Comparison Between Computational and Experimental Results". *AIAA*, 2005. 2005-884.
12. Computational Research and Engineering Acquisition Tools And Environments (CREATE) - Air Vehicles, 201 W Eglin Blvd., Bldg 380, Rm 115 Eglin AFB, FL 32542. *Kestrel User's Guide*, 2 edition.
13. Cummings, Russel M., Scott A. Morton, and David R. McDaniel. "Experiences in Accurately Predicting Time-Dependant Flows". *Progress in Aerospace*, 44(4):241–257, May 2008. Elsevier.

14. Dean, J., S.A. Morton, D.R. McDaniel, and Grtz. S. "Efficient High Resolution Modeling of Fighter Aircraft with Stores for Stability and Control Clearance". *AIAA*, (April), 2007. 2007-1652.

15. Dean, John P., Scott A. Morton, David R. McDaniel, James D. Clifton, and David J. Bodkin. "Aircraft Stability and Control Characteristics Determined by System Identification of CFD Simulations". *AIAA*, 2008. 2008-6378.

16. Forsythe, J. R., C. M. Fremaux, and R. M. Hall. "Calculation of Static and Dynamic Stability Derivatives of the F/A-18E in Abrupt Wing Stall Using RANS and DES". *Proceedings of International Conference for CFD*. 2004. Toronto, Canada.

17. Forsythe, J. R., K. A. Hoffmann, and F. F. Dieteker. "Detached-Eddy Simulation of a Supersonic Axisymmetric Base Flow with an Unstructured Flow Solver". *AIAA*, June 2000. 2000-2410.

18. Forsythe, J. R., K. D. Squires, K. E. Wurtzler, and P. R. Spalart. "Detached-Eddy Simulation of Fighter Aircraft at High Alpha". *Journal of Aircraft*, 41(2):193–200, 2004.

19. Forsythe, J. R. and S. H. Woodson. "Unsteady CFD Calculations of Abrupt Wing Stall using Detached-Eddy Simulation". *AIAA*, Jan 2003. 2003-0594.

20. Fremaux, C. Michael and Robert M. Hall. "COMSAC: Computational Methods for Stability and Control". *NASA/CP-2004-213028/PT1*, April 2004.

21. Gortz, Stefan, David R. McDaniel, and Scott A. Morton. "Towards an Efficient Aircraft Stability and Control Analysis Capability using High-Fidelity CFD". *AIAA*, 2007. AIAA 2007-1053.

22. Haller, G. "An objective definition of a vortex". *Journal of Fluid Mechanics*, 525:1–26, 2005.

23. Jategaonkar, R. V. "Flight Vehicle System Identification - A Time Domain Methodology". *AIAA - Progress in Astronautics and Aeronautics*, 216, 2006.

24. Jategaonkar, R. V., D. Fischenberg, and W. Gruenhagen. "Aerodynamic Modeling and System Identification from Flight Data - Recent Applications at DLR". *Journal of Aircraft*, 41(4):861–91, 2004. 0021-8669.

25. Johnson, Forrester T., Edward N. Tinoco, and N. Jong Yu. "Thirty Years of Development and Application of CFD at Boeing Commercial Airplanes, Seattle". *AIAA*, 2003. AIAA 2003-3439.

26. Klein, Vladislav and Eugene A. Morelli. *Aircraft System Identification: Theory and Practice*. AIAA Education Series, 2006.

27. Laboratory, U.S. Air Force Reseach. "Raptor User Guide", Feb 2012. URL `http://www.arfl.hpc.mil/docs/raptorUserGuide.html`.

28. The Mathworks, Inc. ***Matlab***® *Help Documentation*, r2010b edition, 2010.

29. Morelli, Eugene A. "Practical Input Optimization for Aircraft Parameter Estimation Experiments". *NASA/CR*, May 1993. NASA-CR-191462.

30. Morelli, Eugene A. "Global Nonlinear Aerodynamic Modeling using Multivariate Orthogonal Functions". *Journal of Aircraft*, 32(2):270–77, Mar-Apr 1995.

31. Morelli, Eugene A. "Global Nonlinear Parametric Modeling with Application to F-16 Aerodynamics". *ACC*, June 1998. WP04-2 Paper ID i-98010-2.

32. Morelli, Eugene A. "System Identification Programs for Aircraft (SIDPAC)". *AIAA*, 2002. 2002-4704.

33. Morelli, Eugene A. and Vladislav Klein. "Application of System Identification to Aircraft at NASA Langley Research Center". *Journal of Aircraft*, 42(1):12–25, 2005.

34. Morton, S., S. Goertz, D. McDaniel, and J. Dean. "Computational Aircraft and Armament Stability and Control Techniques Applied to the F-16". Destin, FL, April 2006. ITEA Store Compatibility Symposium.

35. Morton, S. A., M. B. Steenman, R. M. Cummings, and J. R. Forsythe. "DES Grid Resolution Issues for Vortical Flows on a Delta Wing and an F-18C". *AIAA*, Jan 2003. 2003-1103.

36. Morton, S.A., J. Dean, D.R. McDaniel, and S. Grtz. "Poster Paper on Computational Aircraft and Armament Stability and Control Techniques Applied to the F-16". Orlando, FL, November 2006. International Test and Evaluation Association National Symposium.

37. Morton, S.A., S. Grtz, and D.R. McDaniel. "Computational Stability and Control Analysis of the F-16". 2006. Maui High Performance Computing Center Application Briefs.

38. Morton, S.A., D.R. McDaniel, and R.M. Cummings. "F- 16XL Unsteady Simulations for the CAWAPI Facet of RTO Task Group AVT- 113". Reno, NV, Jan 2007. AIAA Paper 2007-0493, Aerospace Sciences Meeting.

39. Morton, Stephen A., J. R. Forsythe, K. D. Squires, and Russel M. Cummings. "Detatched-Eddy Simulations of Full Aircraft Experiencing Massively Separated Flows". *Computational Fluid Dynamics Journal ISCFD Japan*, 13(3), January 2005.

40. Murman, S. M., N. M. Chaderjian, and S. A. Pandya. "Automation of a Navier-Stokes S&C Database Generation for the Harrier in Ground Effect". *AIAA*, 2002. 2002-259.

41. Nichols, R. H. *Turbulence Models and Their Application to Complex Flows.* Technical report, University of Alabama at Birmingham. Revision 4.01.

42. Phillips, Warren F. *Mechanics of Flight.* John Wiley & Sons, Inc, 1 edition, 2004.

43. Pope, Stephen B. *Turbulent Flows.* Cambridge University Press, 2000.

44. Rogers, S. E., M. J. Aftomis, S. A. Pandya, N. M. Chaderjian, E. T. Tenjnil, and J. U. Ahmad. "Automated CFD Parameter Studies on Distributed Parallel Computers". *AIAA*, June 2003. 2003-4229.

45. Spalart, Philippe R. "Young-Person's Guide to Detached-Eddy Simulation Grids". *NASA/CR-2001-211032*, July 2001.

46. Spalart, Philippe R., S. Deck, M. L. Shur, and K. D. Squires. "A New Version of Detached-Eddy Simulation, Resistant to Ambiguous Grid Densities". *Theories of Computational Fluid Dynamics*, May 2006.

47. Versteeg, H K and W Malalasekera. *An Introduction to Computational Fluid Dynamics: The Finite Volume Method.* Pearson Education, 2 edition, 2007.

Vita

Jed Butler graduated from Utah State University with a BS in Mechanical Engineering with an emphasis in Aerospace Engineering. Before attending AFIT, Mr. Butler worked in the Engineering Home Office, Aeronautical Systems Center at Wright-Patterson AFB, OH. There he served as a stability and control engineer. After graduation, Mr. Butler will return to his previous office for a short period, after which he will most likely be sent to the F-15 SA Program Office, also at Wright-Patterson AFB, where he will continue to work as a stability and control engineer.

Permanent address: 2950 Hobson Way
Air Force Institute of Technology
Wright-Patterson AFB, OH 45433

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704–0188*

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704–0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202–4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE (DD–MM–YYYY) 22-03-2012	2. REPORT TYPE Master's Thesis	3. DATES COVERED *(From — To)* Oct 2010 – Mar 2012

4. TITLE AND SUBTITLE Accurate Modeling of Stability & Control Properties for Fighter Aircraft from CFD	5a. CONTRACT NUMBER 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) Jedediah H. Butler	5d. PROJECT NUMBER 5e. TASK NUMBER 5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/ENY) 2950 Hobson Way WPAFB OH 45433-7765	8. PERFORMING ORGANIZATION REPORT NUMBER AFIT/GAE/ENY/12-M04
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Dr. David R. McDaniel AFMC 46 SK/SKI Building 380 Room 115 201 West Eglin Blvd Eglin AFB, FL 32542 850-882-1432 David.McDaniel.ctr@Eglin.af.mil	10. SPONSOR/MONITOR'S ACRONYM(S) DoD HPCMP/CREATE/AV/Kestrel 11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

13. SUPPLEMENTARY NOTES This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

14. ABSTRACT

Difficulties in obtaining accurate Stability and Control (S&C) data for nonlinear regions in the flight envelope early in the design phase often lead to costly fixes late in the acquisition process. Work examined herein addresses this problem by utilizing recent advances in Computational Fluid Dynamics (CFD) to create a high-fidelity database before any parts have to be physically produced. In combination with CFD, System Identification (SID) is used to characterize the S&C characteristics of aircraft by some Reduced Order Model (ROM). The method of obtaining enough data in the "right" places, via some optimized maneuver, to create an accurate ROM is the main focus of this work. A series of metrics were created and validated to help identify the quality of any maneuver before it is even run. To validate the metrics, seven different maneuvers were generated, run, and used to create models. These models are then evaluated against sets of results from validation maneuvers to show accuracy of modeling a given regressor space - the regressor space" being the required variations of the model variables (angle of attack, pitch rate, etc.). From the validated metrics, the "best" maneuver for creating the most accurate model of a given grid within a specified regressor space is defined.

15. SUBJECT TERMS
Stability & Control, S&C, Computational Fluid Dynamics, CFD, Modeling, ROM, Regressor

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON Maj Andrew J. Lofthouse
a. REPORT	b. ABSTRACT	c. THIS PAGE	UU	195	19b. TELEPHONE NUMBER (Include Area Code) (937)674-4043
U	U	U			

Standard Form 298 (Rev. 8–98)
Prescribed by ANSI Std. Z39.18